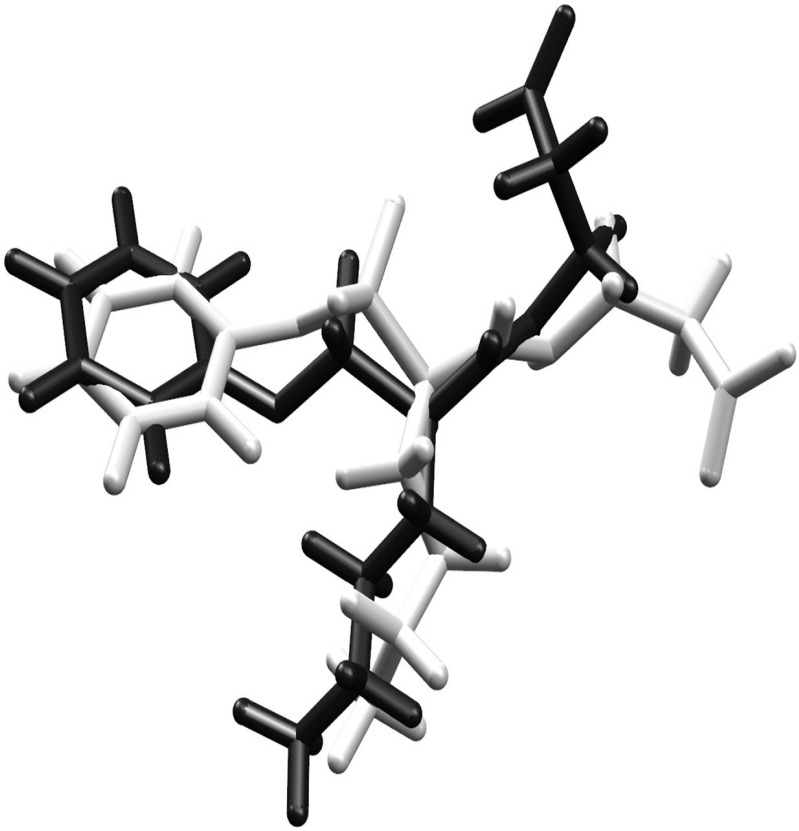
molegro virtual docker

user manual

MVD 2013.6.0 for Windows, Linux, and Mac OS X



copyright CLC bio 2013

Molegro – A CLC bio company

Copyright © 2005–2013 Molegro – A CLC bio company. All rights reserved.

Molegro Virtual Docker (MVD), Molegro Data Modeller (MDM), Molegro Virtual Grid (MVG), and MolDock are trademarks of CLC bio.

All other trademarks mentioned in this user manual are the property of their respective owners.

All trademarks are acknowledged.

Information in this document is subject to change without notice and is provided "as is" with no warranty. CLC bio makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. CLC bio shall not be liable for errors contained herein or for any direct, indirect, special, incidental, or consequential damages in connection with the use of this material.

Table of Contents

1 Introduction to Molegro Virtual Docker..... 7
1.1 Contact Information..... 8
1.2 System Requirements..... 8
1.3 Reporting Program Errors..... 8
1.4 Text Formats Used in the Manual..... 9
1.5 Keyboard Shortcuts..... 9
1.6 Screenshots Used In the Manual..... 9
1.7 Future Updates..... 9
2 Docking Tutorial..... 11
2.1 Importing and Preparing Molecules..... 11
2.2 Running the Docking Simulation..... 17
2.3 Viewing the Results..... 22
3 User Interface..... 24
3.1 Basic Concepts..... 24
3.2 Overview..... 24
3.3 Toolbar..... 25
3.4 Workspace Explorer..... 26
3.5 Properties Window..... 28
3.6 Console Window..... 31
3.7 Clipping Planes..... 32
3.8 Creating a Search Space..... 32
3.9 Hiding Distant Residues..... 33
3.10 Workspace Finder..... 34
3.11 Sequence Viewer..... 35
3.12 Workspace Properties..... 37
3.13 Measurements and Annotations..... 37
3.14 Selection of Atoms, Amino Acids, Rings, and Molecules..... 38
3.15 Custom Coloring of Atoms, Amino Acids, and Molecules..... 38
3.16 Creating Labels..... 39
3.17 Creating Molecular Surfaces..... 40
3.18 Creating Protein Backbone Visualizations..... 42
3.19 Making Screenshots..... 45
3.20 Sidechain Minimization..... 45
3.21 Working With Multiple Receptor Conformations..... 48
3.22 Visualization Settings Dialog..... 50
3.23 High-Quality Rendering..... 56
3.24 Biomolecule Generator..... 58
3.25 Structural Alignment of Proteins..... 60
3.26 Structural Alignment of Small Molecules..... 61
3.27 Macro and Menu Editor..... 61
3.28 PDB and SDF Import Notes..... 63
3.29 Energy Maps..... 64
4 Preparation..... 67

4.1 Import of Molecules....67
4.2 Automatic Preparation....68
4.3 Manual Preparation....71
4.4 Protein Preparation....73
4.5 The Protonation Tab....73
4.6 The Mutate and Optimize Tab....76
4.7 The Settings Tab....77
4.8 Customizing the Protonation Templates....78
5 Data Sources....81
5.1 Data Sources Syntax....82
5.2 Using Data Sources....83
6 Docking Functionality....86
6.1 Cavity Prediction....86
6.2 Constraints....87
6.3 Docking Wizard....91
6.4 Run Docking in Multiple Processes....104
6.5 GPU Screening....106
7 Analyzing the Docking Results....110
7.1 Pose Organizer....110
7.2 Saving Molecules and Solutions Found....117
7.3 Ligand Energy Inspector....118
7.4 Ligand Map (2D Depictions)....127
7.5 Pose Modifier....129
7.6 RMSD Matrix....130
8 Sidechain Flexibility....131
8.1 The Setup Sidechain Flexibility Dialog....132
8.2 Sidechain Flexibility in the Docking Wizard....136
8.3 Sidechain Flexibility and Scripting....137
9 Displaceable Water....139
9.1 Docking with Displaceable Water Molecules....141
9.2 Inspecting Results....143
10 Template Docking....148
10.1 Template Scoring Function....148
10.2 Setting up Template Docking....150
10.3 Docking with Templates....154
10.4 Inspecting Results....155
11 Customizing Molegro Virtual Docker....157
11.1 General Preferences....157
11.2 Command Line Parameters....163
11.3 Changing Re-ranking Score Coefficients....164
12 Obtaining the Best Docking Results....165
12.1 Preparation....165
12.2 Docking....166
12.3 Post-analysis....167
13 Molegro Data Modeller Integration....168

14 Molecular Descriptor Calculations..... 170
14.1 Using the Descriptor Calculation Wizard.....170
14.2 Descriptors in MVD.....171
14.3 Choosing an Output Format.....174
14.4 Working with Molecular Descriptors.....174
14.5 Chemical Feature Distance Matrix Descriptors.....175
15 Molegro Virtual Grid..... 179
15.1 Security Considerations..... 180
15.2 Network and Firewall Issues..... 180
15.3 Licensing..... 181
15.4 Running the Agents.....181
15.5 The Agent GUI..... 182
15.6 Console Mode.....183
15.7 Agent Web Interface.....185
15.8 The Virtual Grid Controller 185
15.9 Combining Results..... 187
15.10 License Management..... 188
15.11 How Virtual Grid Works..... 188
16 Help..... 191
16.1 PDF Help..... 191
16.2 Tip of the Day..... 191
16.3 The Molegro Website 192
16.4 Technical Support..... 192
17 Script Interface..... 193
17.1 Using the Script Interface.....193
17.2 Running a Text-file Script..... 193
17.3 Examples of Common Script Jobs.....194
17.4 Running the Script Interface Interactively.....195
17.5 Running the Script Interface From Python.....196
18 Appendix I: MolDock Scoring Function.....198
19 Appendix II:PLANTS Scoring Function..... 205
20 Appendix III: MolDock Optimizer.....207
21 Appendix IV: Cavity Prediction.....210
22 Appendix V: Clustering Algorithm.....211
23 Appendix VI: Supported File Formats 213
24 Appendix VII: Automatic Preparation..... 215
25 Appendix VIII: Third Party Copyrights.....217
26 Appendix IX: Keyboard Shortcuts.....219
27 Appendix X: Console and Macro Commands.....220
28 Appendix XI: Script Commands 225
28.1 List of Script Commands Available.....226
28.2 Flow Control.....241
29 Appendix XII: MolDock SE.....243
30 Appendix XIII: Iterated Simplex.....246
31 Appendix XIV: Grid-based Scores.....248

32 Appendix XVI: References.....250

1 Introduction to Molegro Virtual Docker

Molegro Virtual Docker (MVD) is an integrated environment for studying and predicting how ligands interact with macromolecules.

The identification of ligand binding modes is done by iteratively evaluating a number of candidate solutions (ligand conformations) and estimating the energy of their interactions with the macromolecule. The highest scoring solutions are returned for further analysis.

MVD requires a three-dimensional structure of both protein and ligand (usually derived from X-ray/NMR experiments or homology modeling). MVD performs flexible ligand docking, so the optimal geometry of the ligand will be determined during the docking.

The preferred way to get started with MVD is:

- Read the remainder of the introduction (Chapter 1)
- Go through the docking tutorial (Chapter 2).
- Read the instructions on how to use the GUI (Chapter 3).

Overall, Chapters 3 to 9 describe various aspects of MVD from importing and preparing molecules to docking and inspecting the docked solutions. Chapter 8 describes docking with flexible sidechains. Chapter 13 describes the Molegro Data Modeller integration, which makes it possible to perform advanced data visualization and modelling. Chapter 17 provides an overview of the scripting features in MVD.

More detailed information about the algorithms (cavity detection, clustering, binding mode prediction) and scoring functions (MolDock Score and PLANTS Score) used by MVD can be found in the appendices.

1.1 Contact Information

Molegro Virtual Docker is developed by:

CLC bio – a Qiagen company

Silkeborgvej 2, Prismet

8000 Aarhus

Denmark

http://www.clcbio.com

VAT no.: DK 28 30 50 87

Telephone: +45 70 22 55 09

Fax: +45 70 22 55 19

E-mail: **info@clcbio.com**

If you have questions or comments regarding the program, you are welcome to contact our support function:

E-mail: **support@clcbio.com**

1.2 System Requirements

The system requirements for Molegro Virtual Docker are:

- Windows 7, Vista, 2003, XP, or 2000.
- Linux: Linux: The officially supported system is Ubuntu 12.04 LTS. We provide both 32 and 64 bit builds. Please send a mail to support@clcbio.com if you experience problems running the software on your system.
- Mac OS X 10.6 Intel (and later versions).

1.3 Reporting Program Errors

If you discover a program error, please mail the information to:

support@clcbio.com

Remember to specify how the error can be reproduced, the version number of Molegro Virtual Docker in question, and the operating system that was used. If possible, inclusion of molecular files used (e.g. Mol2, PDB, MVDML) will make it easier for us to reproduce (and correct) the error.

1.4 Text Formats Used in the Manual

The following formatting styles are used in this manual:

- All GUI text, labels, and keyboard shortcuts are written in bold face with initial capital letters.

  Examples: **Workspace Explorer**, **Macro Definition**, **Ctrl-O**

- Menus and menu items are identified using dividing lines and bold face.

  Example: **View** | **Docking View** indicates that the user should first select the **View** menu and then select the **Docking View** menu item.

- Filenames are written in mono-spaced font.

  Example: `\Molegro\MVD\bin\mvd.exe`

1.5 Keyboard Shortcuts

The keyboard shortcuts used in the manual works for Windows and Linux versions of MVD. On Mac OS X, the **CTRL** key is replaced by the **command** key and function key shortcuts (e.g. **F1**) should be invoked by pressing the function key and the **fn** key (e.g. **fn+F1**).

1.6 Screenshots Used In the Manual

The screenshots used in the manual are taken from the Windows XP and Vista versions of MVD. Therefore, dialogs and other GUI related material may slightly differ on Linux and Mac OS X versions.

1.7 Future Updates

Molegro Virtual Docker contains a built-in version checker making it easy to check for new program updates including new features and bug fixes. To check for new updates, select **Help** | **Check for Updates**. A window showing available updates and details about changes made will appear (see Figure 1).

New Version Found
Latest version available
2010.4.1.0
Your current version
2009.3.2.1
Changes:
June 9th 2010: MVD Version 4.1.0
Molegro Virtual Docker 4.1.0 updates the Data Analyzer and improves Molegro Virtual Grid performance and stability.
Notice that it is possible (and recommended) to update trial licenses as long as the trial license key is still valid.
February 10th 2010: MVD Version 4.0.2
Molegro Virtual Docker 4.0.2 is a minor update, which adds better support for very large molecules, foreign (unicode) characters in MVG scripts, and fixes a few bugs.
A new version has been found. Please go to www.molegro.com to update your application.
OK

Figure 1: Check for updates.

2 Docking Tutorial

This tutorial will go through a simple docking exercise by redocking a co-crystallized ligand to its native binding site. The tutorial will highlight aspects such as import and preparation of molecules, conducting the actual docking run, and visual inspection of the poses found.

2.1 Importing and Preparing Molecules

In order for MVD to be able to perform optimally, the molecules in the workspace must be properly prepared before the docking begins. The molecules can either be prepared internally in MVD, or externally by another program (e.g. MOE from CCG [CCG] or Maestro from Schrödinger, LLC [SCHRODINGER]). In this tutorial we will use the built-in preparation method available in MVD.

File Import

If the workspace is not empty, start by clearing it (select **File** | **Clear Workspace**). Next we will add some structures. This can be accomplished by selecting **File** | **Import Molecules...** or by dragging and dropping a molecule structure file. MVD supports PDB, Mol2, SDF, and its own XML-based format, MVDML.

Start by importing the file `1HVR.pdb` from the installation `examples` directory (located in the MVD installation folder). This file (a HIV-1 protease complexed with XK263) is an unmodified file taken from the RCSB Protein Data Bank (www.pdb.org).

Choosing Molecules to Import

The **Import Molecules** dialog (see Figure 2) appears.

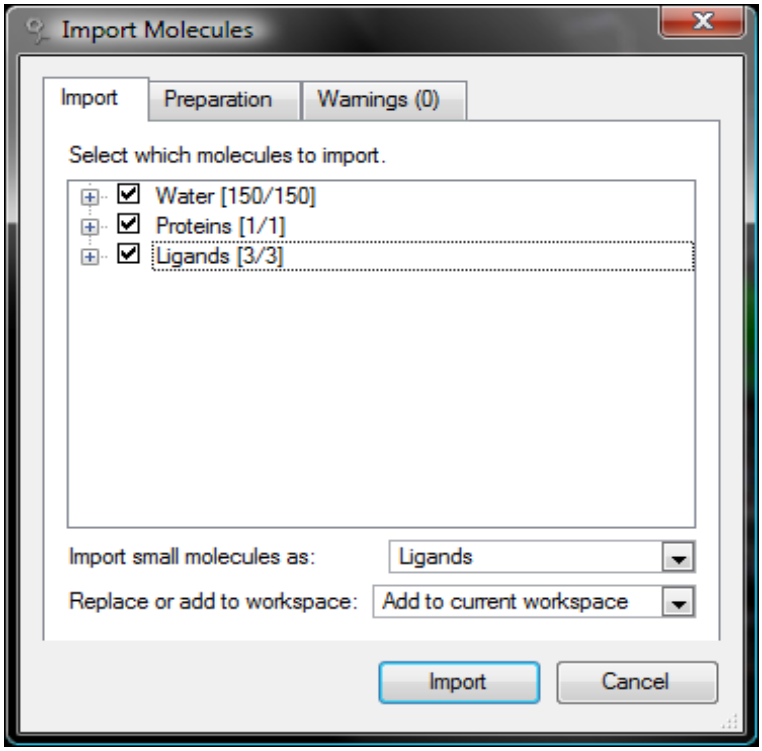
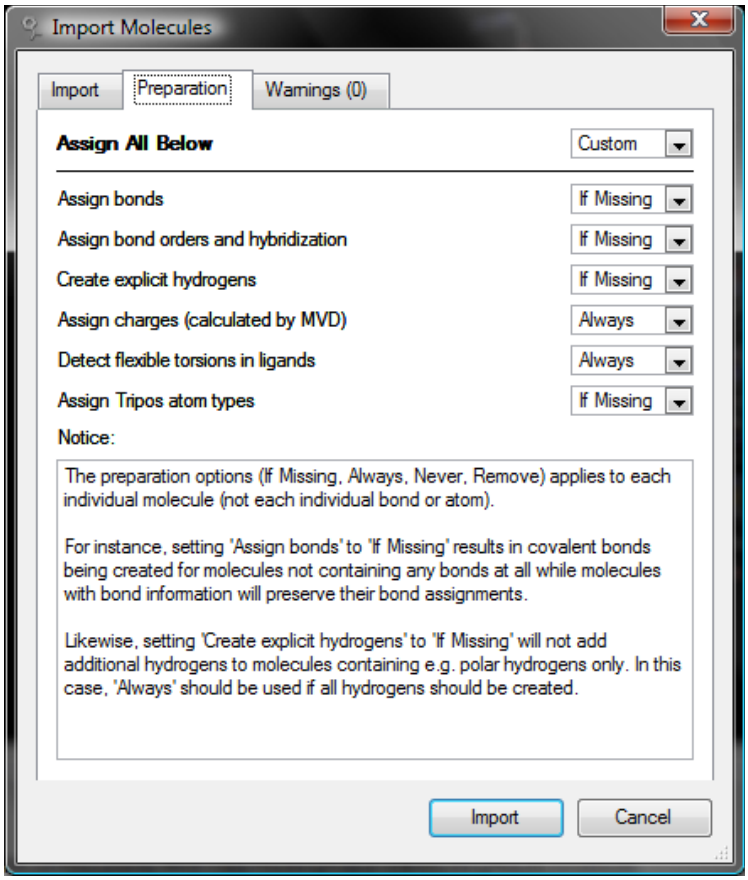


Figure 2: Importing 1HVR from the PDB file.

Deselect the cofactors since we will not need these for this example. The import dialog shows two proteins: actually these are two chains from the same protein. It also indicates that a ligand has been detected in the PDB file.

Choosing Preparation Types

Select the **Preparation** tab (see Figure 3). Some structures contain information about bond types and bond orders, and have explicit hydrogens assigned. However, PDB files often have poor or missing assignment of explicit hydrogens, and the PDB file format cannot accommodate bond order information. Set **Assign All Below** to **Always**. This ensures that all preparation will be done by MVD.


Import Molecules
Import
Preparation
Warnings (0)
Assign All Below
Custom
Assign bonds
If Missing
Assign bond orders and hybridization
If Missing
Create explicit hydrogens
If Missing
Assign charges (calculated by MVD)
Always
Detect flexible torsions in ligands
Always
Assign Tripos atom types
If Missing
Notice:
The preparation options (If Missing, Always, Never, Remove) applies to each individual molecule (not each individual bond or atom).
For instance, setting 'Assign bonds' to 'If Missing' results in covalent bonds being created for molecules not containing any bonds at all while molecules with bond information will preserve their bond assignments.
Likewise, setting 'Create explicit hydrogens' to 'If Missing' will not add additional hydrogens to molecules containing e.g. polar hydrogens only. In this case, 'Always' should be used if all hydrogens should be created.
Import
Cancel
Figure 3: Preparing the PDB-file.

If the protein structure has been prepared beforehand and saved in a format capable of handling all structural information (e.g. Mol2 file format), you should import it via the default preparation setting **If Missing**. This setting only performs a given preparation if the required information cannot be found in the file.

Notice: Per default any charge information for a molecule is ignored ('Assign charges = Always' means that MVD's internal charge scheme is always used to calculate charges). If you want to use partial charges stored with the molecule, set 'Assign charges = If Missing' – this way MVD will only calculate charges for molecules with no charge information present. If charges are imported from molecules (e.g. provided with Mol2 files), partial charges assigned to hydrogens will be moved to bonded heavy-atom since explicit hydrogens are not taken into account by the scoring functions used during docking.

Inspecting the Warnings

The last tab in the import dialog (**Warnings (0)**) shows potential problems with the structure file. In this case no warnings are reported.

Now click the **Import** button. The protein and the ligand appears in the **Visualization Window** (see Figure 4).

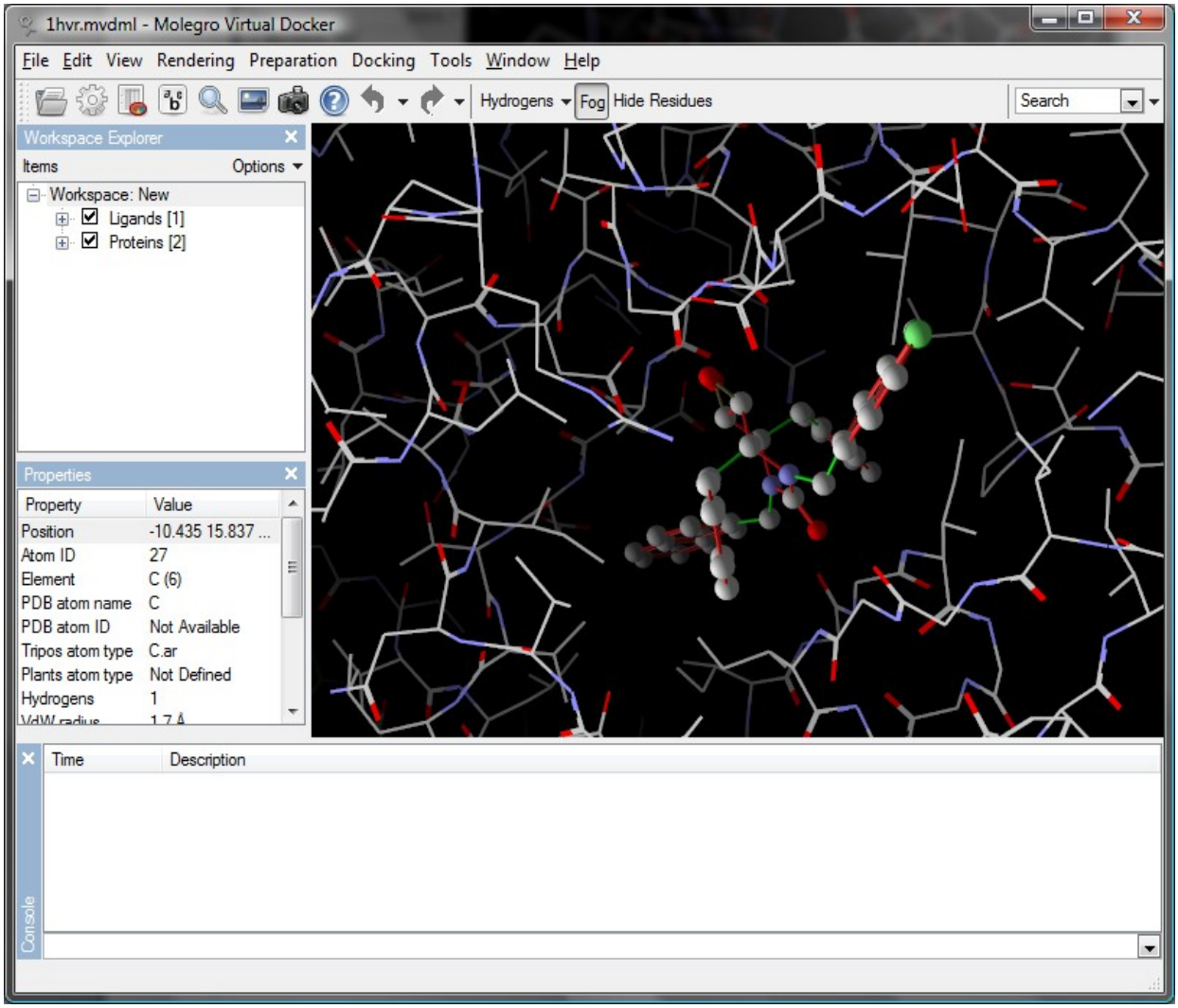


Figure 4: The imported structure.

In order to inspect the imported ligand, hide the protein by clicking on the check box next to the **Proteins** category in the **Workspace Explorer** (see Figure 5).

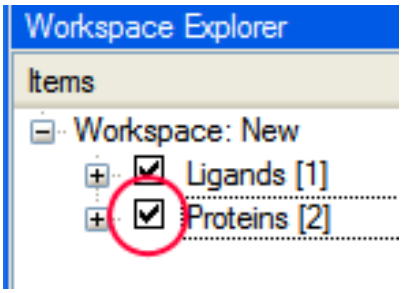


Figure 5: Hiding the Protein.

Now zoom in on the ligand (see Figure 6).

Zooming can be performed using either:

- the scroll wheel on the mouse
- by pressing and holding both mouse buttons
- by pressing shift and holding left mouse button

It is also possible to choose **Fit to Screen** from the context menu for ligands in the **Workspace Explorer**. Notice that the ligand has been assigned bond orders, aromatic rings have been detected, and explicit hydrogens have been added. Also notice that some bonds are green. These bonds will be set flexible during the docking simulation. If a bond should be held rigid during the simulation, right-click on it, and choose **Set Flexibility** from the context menu.

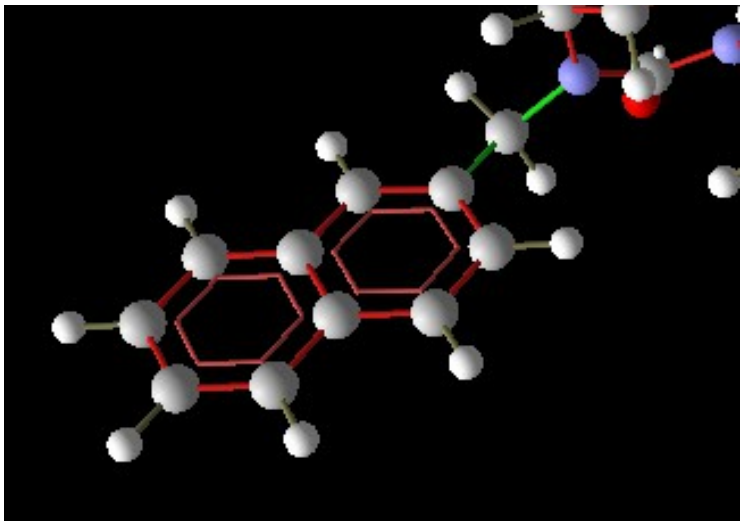


Figure 6: Inspecting flexible bonds.

Adding a Molecular Surface

Next we will add a surface to get an impression of the structure of the protein. We will do this by choosing **Create Surface...** from the **Proteins** context menu in the **Workspace Explorer** (see Figure 7).

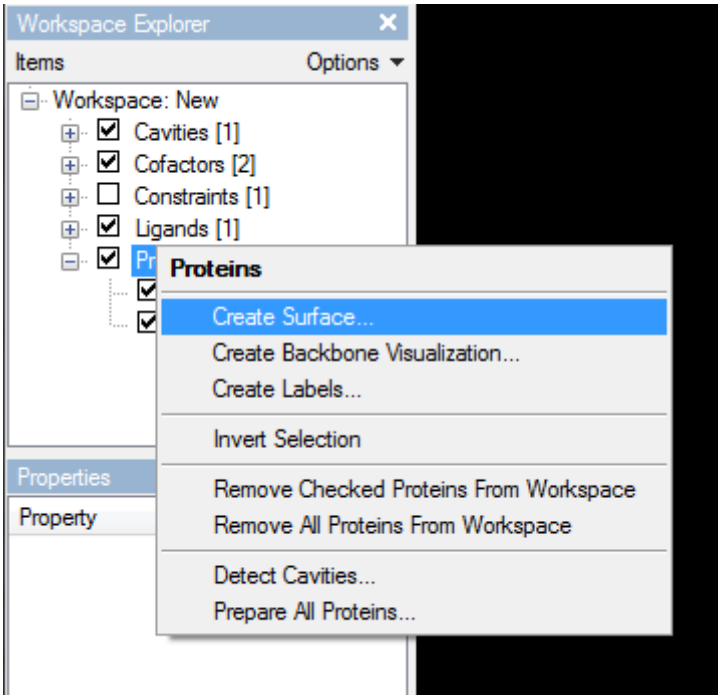


Figure 7: Surface creation.

In the dialog that appears, just click **OK**. This will create a protein surface based on the default settings which are an opaque solvent accessible surface colored according to the electrostatic potential (red and blue colored areas correspond to regions with respectively negatively and positively charged residues). Notice that the surface also show up as an element in the **Workspace Explorer** (**Surfaces** category).

Predicting the Binding Site

Next we will try to narrow down the potential binding site for the protein. This can be done automatically by selecting **Preparation** | **Detect Cavities**. After pressing the **OK** button, the system will predict a binding site in the center of the protein (see Figure 8) using the algorithm described in Appendix IV: Cavity Prediction.

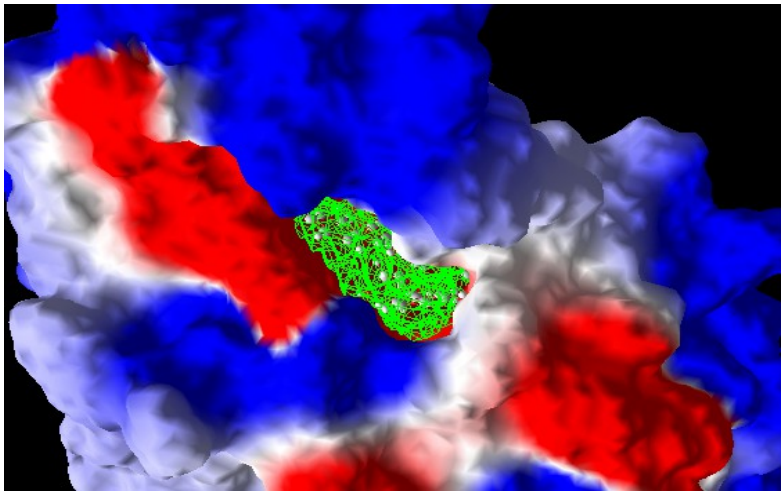


Figure 8: The predicted binding site.

Now we are ready to start the docking. To get a clearer view of this process start by selecting **View** | **Reset View!**. This will reset the 3D view (and hide the surface). Now select **View** | **Docking View**. This will switch to a view where ligands and poses have different colors and capped-stick representation instead of ball-and-stick.

2.2 Running the Docking Simulation

The **Docking Wizard** is invoked by selecting **Docking** | **Docking Wizard**.

Choosing Structures

The first tab shows which structures are included in the simulation. If multiple ligands are available they can be chosen here. Since we are doing a redocking study here, we will use the only available ligand as reference: Set **Reference Ligand** to **XK2_263** and continue to the next tab by pressing **Next**.

Defining the Region of Interest

The most important thing on the next tab is to set the binding site. Since we have detected cavities, we set **Origin** to **Cavity 1...**. If the protein had multiple potential binding sites, more choices would appear (see Figure 9).

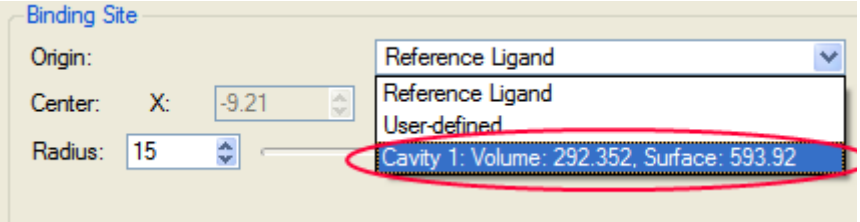


Figure 9: Selecting the binding site.

Now continue to the next tab where **Search Parameters** can be set. We will not change any search parameters – press the **Next** button to proceed to the next tab.

The next tab (**Pose Clustering**) allows you to configure whether multiple poses should be returned. We will stick to the default setting, which will limit the number of poses returned to a maximum of five. Continue to the next tab.

The **Errors and Warnings** tab in the **Docking Wizard** shows potential problems with the docking setup (if any). It should not show any warnings at this stage. Press the **Next** button to proceed to the last tab.

In the **Setup Docking Execution** tab (see Figure 10), several choices are available for executing the docking simulation. We will use the default settings (the settings are further explained in Section 6.3). Finally, the **Output directory** specifies where the docking data (log file and found poses) will be stored. Choose a directory pressing the "**...**" button or keep the default settings.

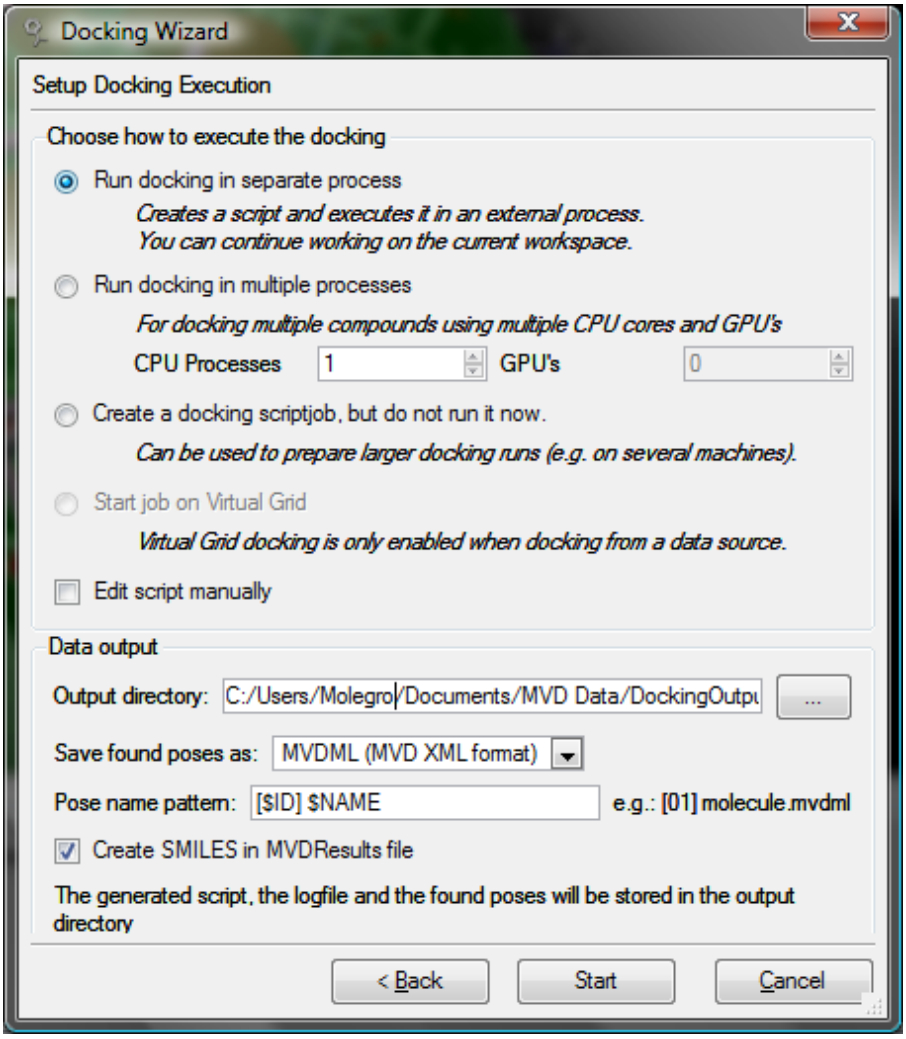

Docking Wizard
Setup Docking Execution
Choose how to execute the docking
Run docking in separate process
Creates a script and executes it in an external process.
You can continue working on the current workspace.
Run docking in multiple processes
For docking multiple compounds using multiple CPU cores and GPU's
CPU Processes
1
GPU's
0
Create a docking scriptjob, but do not run it now.
Can be used to prepare larger docking runs (e.g. on several machines).
Start job on Virtual Grid
Virtual Grid docking is only enabled when docking from a data source.
Edit script manually
Data output
Output directory:
C:/Users/Molegro/Documents/MVD Data/DockingOutpu
...
Save found poses as:
MVDML (MVD XML format)
Pose name pattern:
[$ID] $NAME
e.g.: [01] molecule.mvdml
Create SMILES in MVDResults file
The generated script, the logfile and the found poses will be stored in the output directory
< Back
Start
Cancel

Figure 10: The Batchjob dialog.

Now we can begin the docking simulation by pressing the **Start** button.

The **Molegro Virtual Docker Batchjob** dialog appears showing the docking progress (see Figure 11).

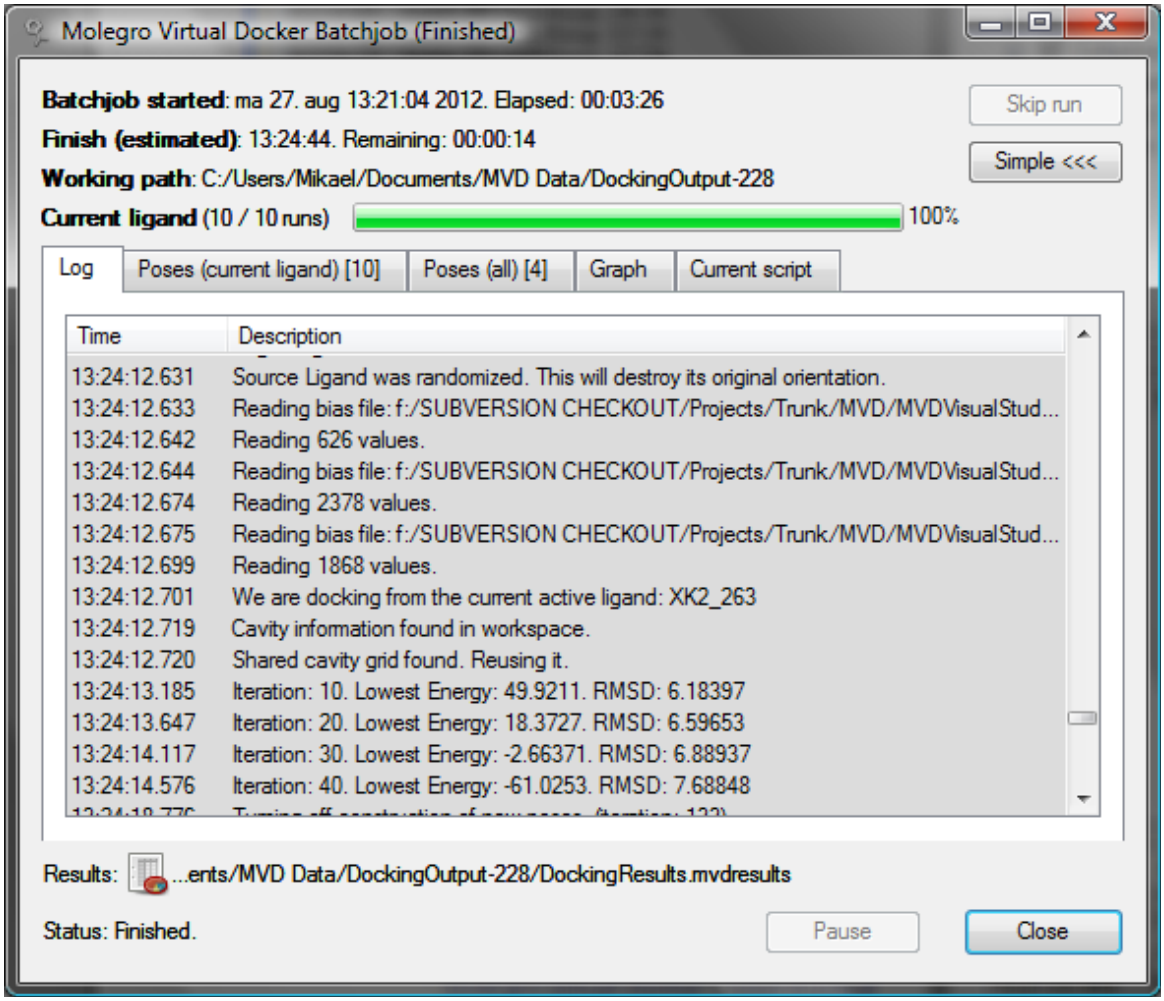

Molegro Virtual Docker Batchjob (Finished)
Batchjob started: ma 27. aug 13:21:04 2012. Elapsed: 00:03:26
Finish (estimated): 13:24:44. Remaining: 00:00:14
Working path: C:/Users/Mikael/Documents/MVD Data/DockingOutput-228
Current ligand (10 / 10 runs)
100%
Skip run
Simple <<<
Log
Poses (current ligand) [10]
Poses (all) [4]
Graph
Current script
Time
Description
13:24:12.631 Source Ligand was randomized. This will destroy its original orientation.
13:24:12.633 Reading bias file: f:/SUBVERSION CHECKOUT/Projects/Trunk/MVD/MVDVisualStud...
13:24:12.642 Reading 626 values.
13:24:12.644 Reading bias file: f:/SUBVERSION CHECKOUT/Projects/Trunk/MVD/MVDVisualStud...
13:24:12.674 Reading 2378 values.
13:24:12.675 Reading bias file: f:/SUBVERSION CHECKOUT/Projects/Trunk/MVD/MVDVisualStud...
13:24:12.699 Reading 1868 values.
13:24:12.701 We are docking from the current active ligand: XK2_263
13:24:12.719 Cavity information found in workspace.
13:24:12.720 Shared cavity grid found. Reusing it.
13:24:13.185 Iteration: 10. Lowest Energy: 49.9211. RMSD: 6.18397
13:24:13.647 Iteration: 20. Lowest Energy: 18.3727. RMSD: 6.59653
13:24:14.117 Iteration: 30. Lowest Energy: -2.66371. RMSD: 6.88937
13:24:14.576 Iteration: 40. Lowest Energy: -61.0253. RMSD: 7.68848
Results: ...ents/MVD Data/DockingOutput-228/DockingResults.mvdresults
Status: Finished.
Pause
Close

Figure 11: Docking Progress dialog.

While the simulation is running the energy of the currently best found pose (the pose with the lowest energy) can be observed on the **Graph** tab page (see Figure 12). The graph shows the docking score (in arbitrary units) as a function of number of iterations performed by the docking search algorithm.

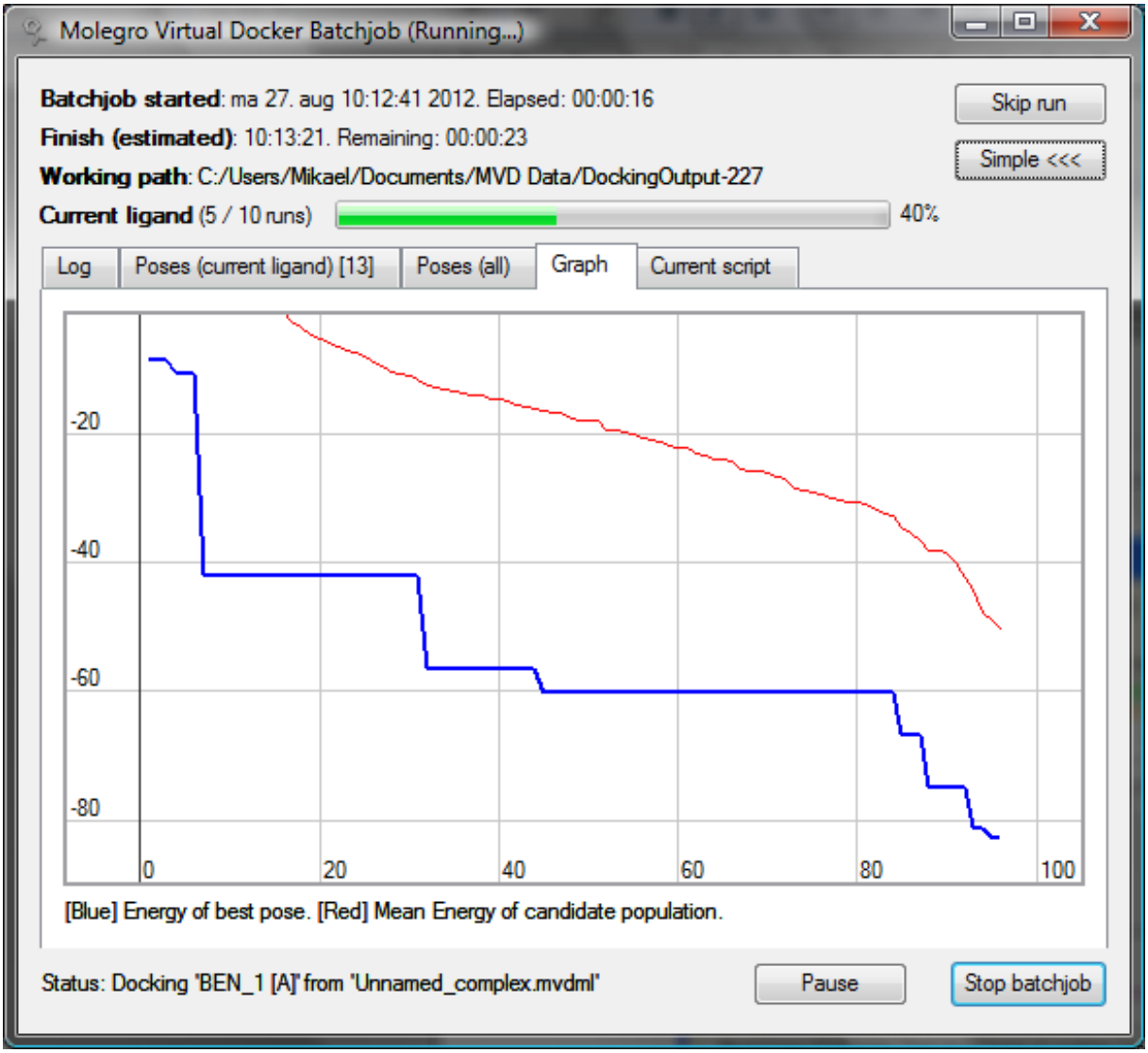


Figure 12: Docking search progress.

Let the simulation run for a while (1-2 minutes or so). The docking engine should find a good solution within 800 iterations. The simulation will eventually time out on its own (after 2,000 iterations or about 100,000 evaluations) or if the simulation has converged.

Notice: Because of the stochastic nature of the docking engine more than one docking run may be needed to identify the correct binding mode.

The docking run can also be stopped by pressing the **Stop batchjob** button. When the docking run finishes, the poses found are saved to the **Output directory** specified previously in the **Docking Wizard** dialog (here *c:\Program Files\Molegro\MVD\ScriptOutput* was used).

The poses found can now be imported into MVD by:

1) Selecting **Import Docking Results (*.mvdresults)...** from the **File** menu using the *DockingResults.mvdresults* file.

2) Dragging and dropping the *DockingResults.mvdresults* file onto the MVD application.
3) Dragging and dropping the *DockingResults* icon onto the MVD application.

The *DockingResults.mvdresults* file is located in the **Output directory** together with a docking log file and the poses found (in Mol2 file format).

After importing the *DockingResults.mvdresults* file, the **Pose Organizer** will appear showing the poses found (see Figure 13).

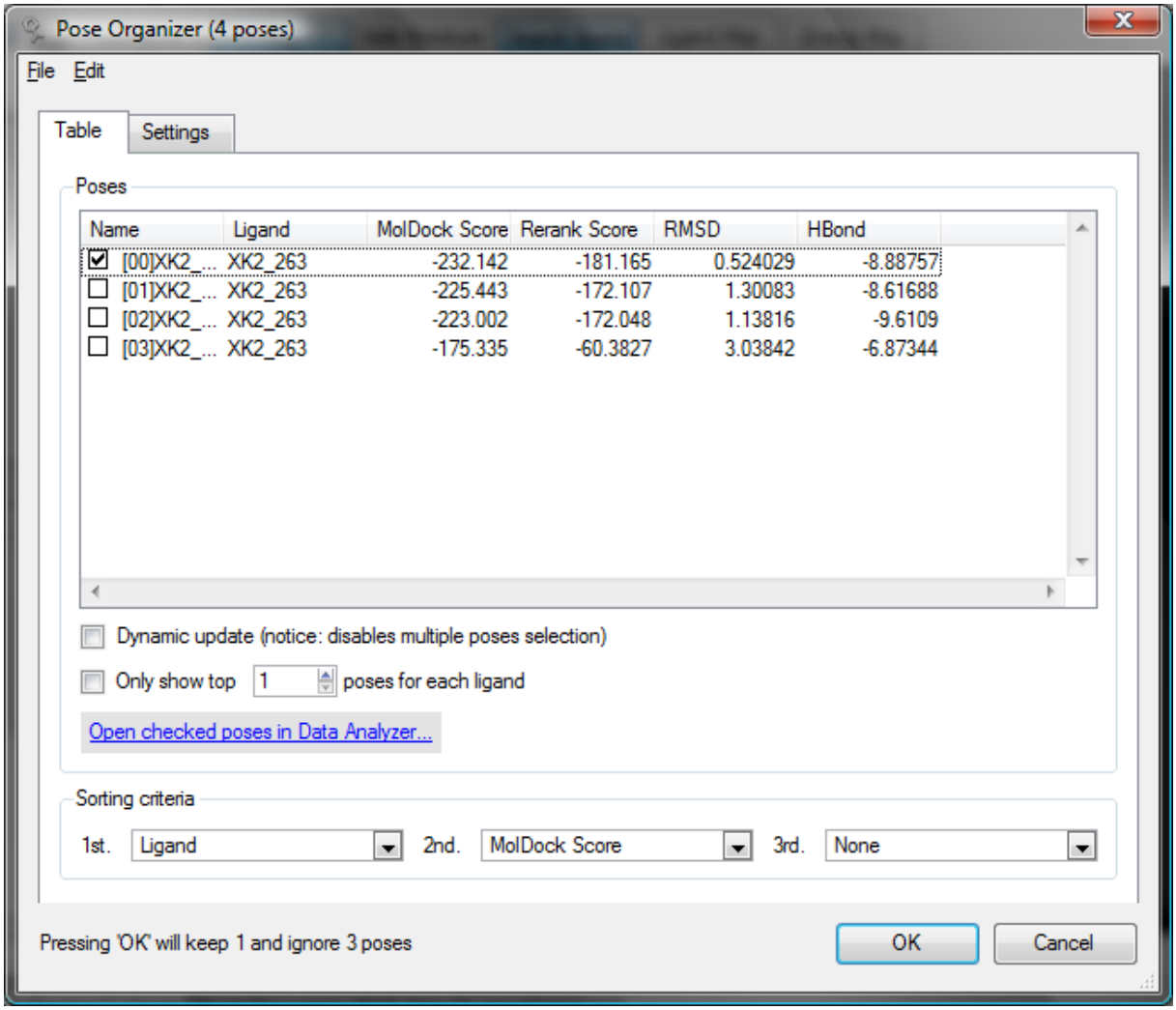

Pose Organizer (4 poses)
File Edit
Table Settings
Poses
Name Ligand MolDock Score Rerank Score RMSD HBond
[00]XK2_... XK2_263 -232.142 -181.165 0.524029 -8.88757
[01]XK2_... XK2_263 -225.443 -172.107 1.30083 -8.61688
[02]XK2_... XK2_263 -223.002 -172.048 1.13816 -9.6109
[03]XK2_... XK2_263 -175.335 -60.3827 3.03842 -6.87344
Dynamic update (notice: disables multiple poses selection)
Only show top 1 poses for each ligand
Open checked poses in Data Analyzer...
Sorting criteria
1st. Ligand 2nd. MolDock Score 3rd. None
Pressing 'OK' will keep 1 and ignore 3 poses
OK
Cancel

Figure 13: The predicted poses.

The **Pose Organizer** allows you to inspect the poses and select which structures to keep (by toggling the select box next to them). At this point we will just add all found poses to the workspace. First select all poses by manually checking them or use the **Edit** | **Check All** menu. Afterwards, press the **OK** button.

2.3 Viewing the Results

At this point it would be a good idea to save the workspace with the new poses added. This can be done by selecting **File** | **Save Workspace...**. This will save the workspace (proteins, ligands, poses, etc.) in MVD's own XML-based format. In order to export the poses to other formats, the **Pose Organizer** can be used.

Revisiting the Pose Organizer

First switch to the pose organizer view (**View** | **Pose Organizer View**). Each pose is shown in different colors. Next open the **Pose Organizer** (select **Docking** | **Pose Organizer**).

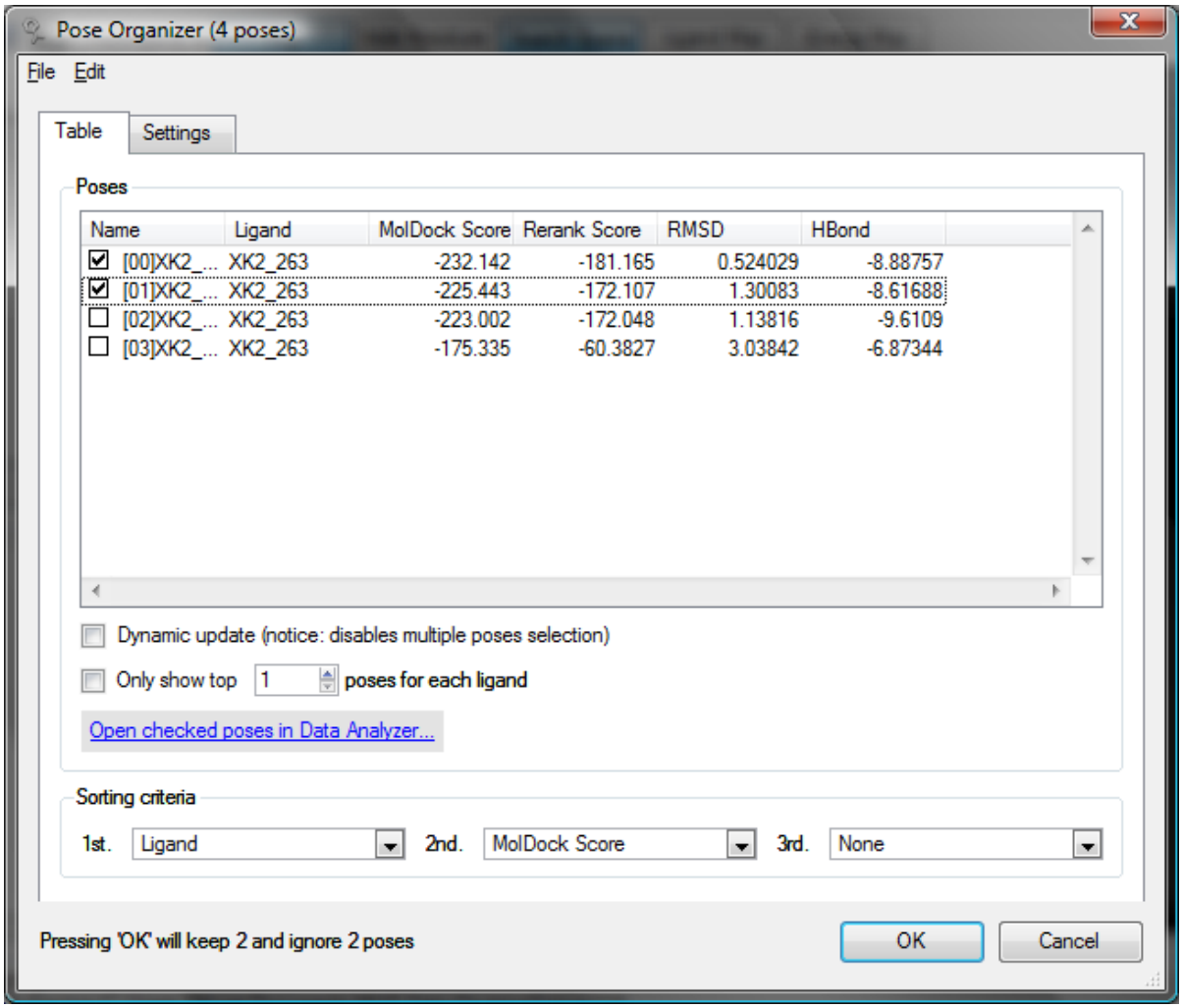

Pose Organizer (4 poses)
File Edit
Table Settings
Poses
Name Ligand MolDock Score Rerank Score RMSD HBond
[00]XK2_... XK2_263 -232.142 -181.165 0.524029 -8.88757
[01]XK2_... XK2_263 -225.443 -172.107 1.30083 -8.61688
[02]XK2_... XK2_263 -223.002 -172.048 1.13816 -9.6109
[03]XK2_... XK2_263 -175.335 -60.3827 3.03842 -6.87344
Dynamic update (notice: disables multiple poses selection)
Only show top 1 poses for each ligand
Open checked poses in Data Analyzer...
Sorting criteria
1st. Ligand 2nd. MolDock Score 3rd. None
Pressing 'OK' will keep 2 and ignore 2 poses
OK
Cancel

Figure 14: Viewing multiple poses.

Turn off the original ligand and the search space sphere (colored green) in the 3D view window (by clicking the **Ligands** and **Constraints** check boxes in the **Workspace Explorer**). The poses in the **Pose Organizer** can be visualized by selecting them (see Figure 14).

By enabling the **Dynamic update** option, we can inspect the individual poses one at a time (single pose view mode). Click on the poses on the list to have them visualized (see Figure 15). Notice that hydrogen bonds are dynamically updated and shown when switching to a new pose.

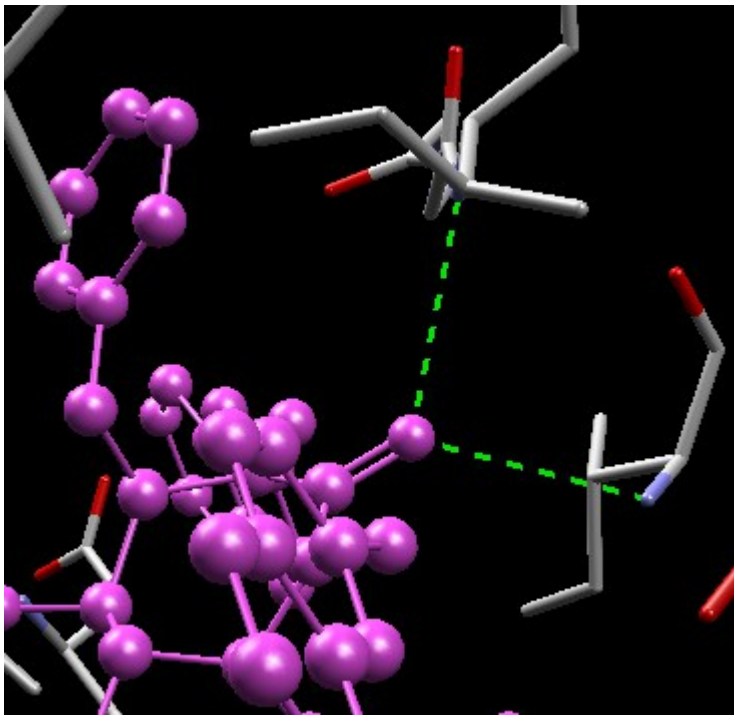


Figure 15: Viewing hydrogen bonds.

The **Pose Organizer** can also automatically rotate rotatable hydrogens (like hydroxyl rotors) in both the receptor and the ligand to their optimal position. It can also be used to rerank the ligands (using a rank score), or view their energy contributions split up into different categories (see Section 7.1 for more details).

This concludes the tutorial.

3 User Interface

3.1 Basic Concepts

Molegro Virtual Docker is based on the notion of workspaces. The *workspace* is the central component and represents all the information available to the user in terms of molecules (proteins, ligands, cofactors, water molecules, and poses), user-defined constraints (visualized as small spheres), cavities (visualized as a grid mesh), and various graphical objects (molecular surfaces, backbone visualizations, labels, etc.).

By default, an empty workspace is shown when starting MVD. A workspace can be saved, cleared, replaced by or appended to other workspaces. The content of the current workspace is listed in the **Workspace Explorer** window, which also allows for manipulation of the various items available (see Section 3.4 for more details).

Notice: When saving a workspace in the internal MVDML format not all 3D visualization objects are saved (e.g. labels, interactions, annotations, backbones, and surfaces). For more information about the MVDML format see Appendix VI: Supported File Formats and Section 7.2.

3.2 Overview

The user interface in MVD is composed of a central 3D view (referred to as the **Visualization Window** or 3D world, together with a number of dockable windows (introduced below).

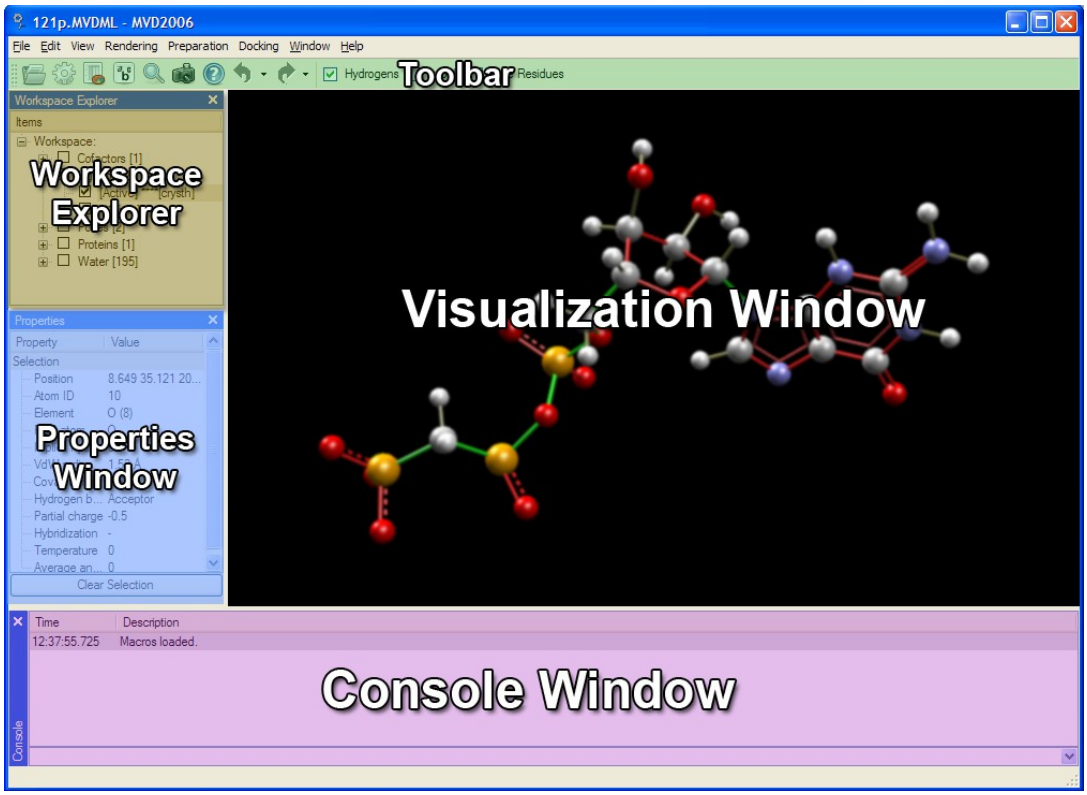


Figure 16: Main application window.

3.3 Toolbar

The **MVD Toolbar** provides easy and fast access to commonly used actions, such as import of molecules, docking using the **Docking Wizard**, and pose inspection using the **Pose Organizer**.

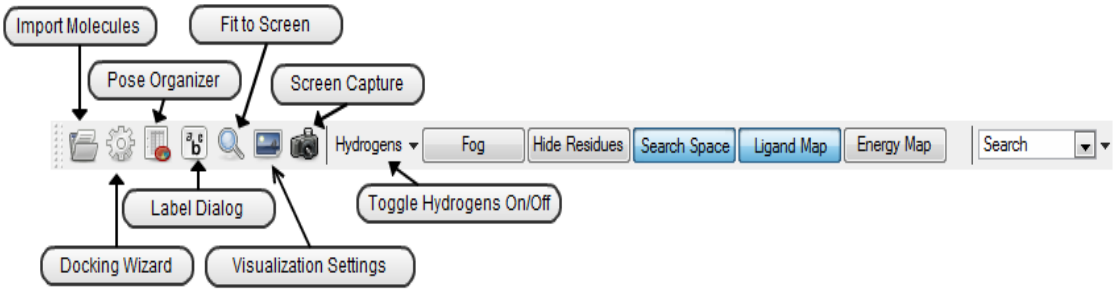


Figure 17: MVD Toolbar.

The **MVD Toolbar** also contains different toggle buttons. The **Hydrogens** button makes it easy to switch between different view modes (**Show all hydrogens**, **Show only polar hydrogens**, and **Hide all hydrogens**). The **Fog** button is used to toggle fog effects on and off. The **Hide Residues** button is used to toggle whether residues should be hidden or not (see Section 3.9 for

more details). The Search Space button makes it possible to define and toggle Search Spaces on and off (see Chapter 3.8) and the **Ligand Map** button is used to toggle on 2D visualization of a ligand or pose and its interactions with the protein (see Section 7.4 for more details). Finally the **Energy Map** button visualizes the force field generated by the proteins in the workspace (see Chapter 3.29). The **Workspace Finder** located at the far right side of the toolbar can be used to quickly search for molecule names and residue/atom IDs (see Section 3.10 for more details).

3.4 Workspace Explorer

The **Workspace Explorer** window (see Figure 18) contains information about the 3D-objects (both molecules, such as proteins, ligands, and water molecules - but also objects such as labels, surfaces, backbones, and cavities).

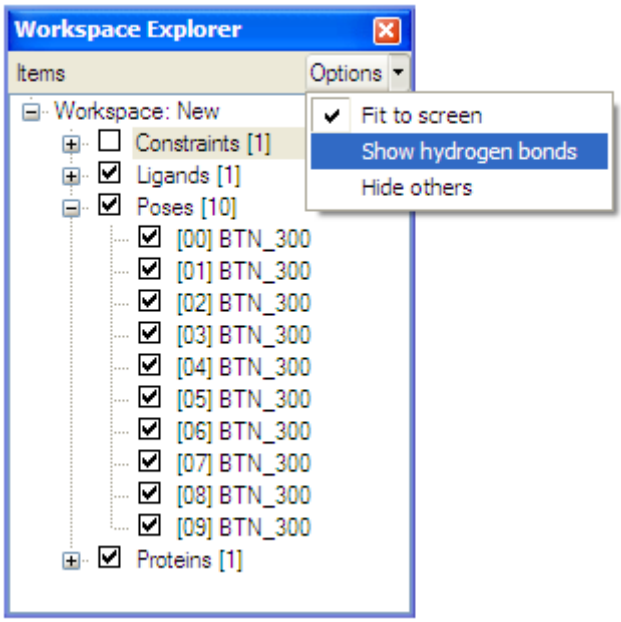


Figure 18: Workspace Explorer window.

The context menu (right mouse button click) allows the user to:

- Export molecules to PDB, Mol2, or SDF format
- Edit workspace properties (workspace title and workspace notes)
- Rename molecules
- Remove items from the current workspace
- Set the currently active ligand
- Set all torsions in a ligand either rigid or flexible
- Copy ligands to poses (used to inspect ligands with the **Pose Organizer**)

- Clone ligand or protein (makes a copy of the molecule)
- Convert ligand to pose or cofactor
- Convert protein to ligand
- Convert pose to ligand (used when docking poses)
- Modify ligand or pose (using the **Pose Modifier**)
- Detect cavities (using the **Cavity Prediction** dialog) and merge them.
- Export cavity grid points to PDB or Mol2 format (represented as water molecules)
- Inspect poses (using the **Pose Organizer**)
- Prepare molecules
- Create labels, surfaces, and backbones
- Fit the molecule to the visualization window

Inspecting Molecules

The **Workspace Explorer** can also be used to inspect molecules in the **Visualization Window** using the left mouse button to select the molecules or by using keyboard shortcuts (see below).

The **Options** button (see Figure 18) contains settings used to customize the behavior when inspecting molecules. The **Fit to screen** option will automatically zoom selected molecules so that they fit into the **Visualization Window**. The **Show hydrogen bonds** option can be used to display hydrogen bonds (only applicable for ligands and poses). The **Hide others** option toggles whether other checked molecules in the current workspace category are allowed or not.

Keyboard shortcuts are also available for inspecting molecules. Pressing the **Shift** button while clicking the left-mouse button on a molecule in the chosen category (e.g. Ligands or Poses) will fit the selected molecule in the **Visualization Window** and all other molecules located in the same category are hidden.

Alternatively, using **Ctrl+Shift** when clicking on a molecule, hydrogen bonds are shown for the selected molecule.

Instead of using the mouse to select molecules to inspect, **Up** or **Down** keys can be used to browse the molecules present in the currently selected **Workspace Explorer** category. If the **Ctrl** and **Shift** shortcuts are omitted, the settings enabled in the **Options** panel will be used.

If multiple receptor conformations are available in the workspace, a drop-down box will appear at the bottom of the **Workspace Explorer** allowing the user to

change between conformations. For more information about working with multiple receptor conformations, see Chapter 8.

3.5 Properties Window

The **Properties Window** contains information about the currently selected (or highlighted) 3D object(s) in the **Visualization Window** and provides useful information while preparing and modifying the molecules.

Figure 19 shows an example of different properties for a highlighted atom.

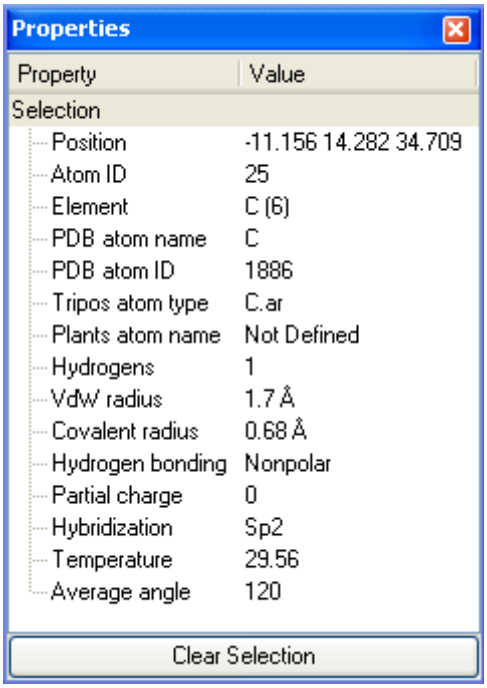


Figure 19: Example of properties for a selected atom.

Visualization Window

The **Visualization Window** (see Figure 20) visualizes all the selected molecules in the workspace and all custom graphical objects (e.g. labels, annotations, charges, protonation guides, backbones, surfaces, and cavities).

Notice: For large molecules it can be computationally slow to display all atoms. Therefore it is recommended to adjust the view to the user's needs. Often it is a good idea to add a molecular surface (perhaps transparent) to give some idea of the 3D structure. Alternatively, switching to wireframe visualization style and hiding non-polar (or all) hydrogens atoms can also improve the visualization speed significantly. Also consider cropping (removing) non-relevant parts of the complex, in order to make the visualization faster. Cropping is described in Section 3.9.

Changing the 3D World Appearance

The visualization engine is highly configurable.

Molecules can be drawn as lines (wireframe), ball-and-sticks, capped-sticks, and space-fill (CPK).

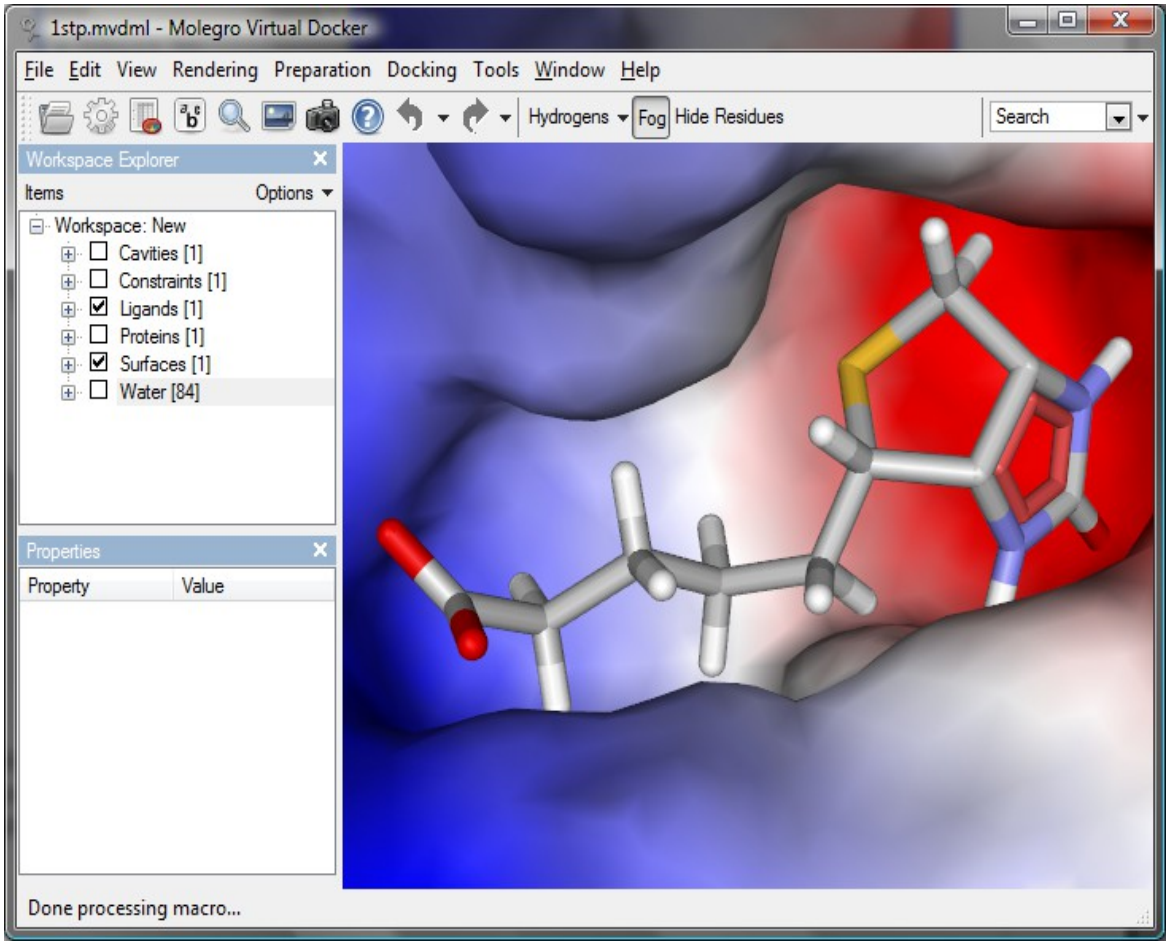


Figure 20: Visualization of Biotin (1STP) in capped-stick style and electrostatic protein surface.

Notice: Ball-and-stick is the preferred style for handling preparation of ligands, since the visualized bond shows bond order, and is color coded to display whether the bond is set rigid (brown or red) or flexible (green).

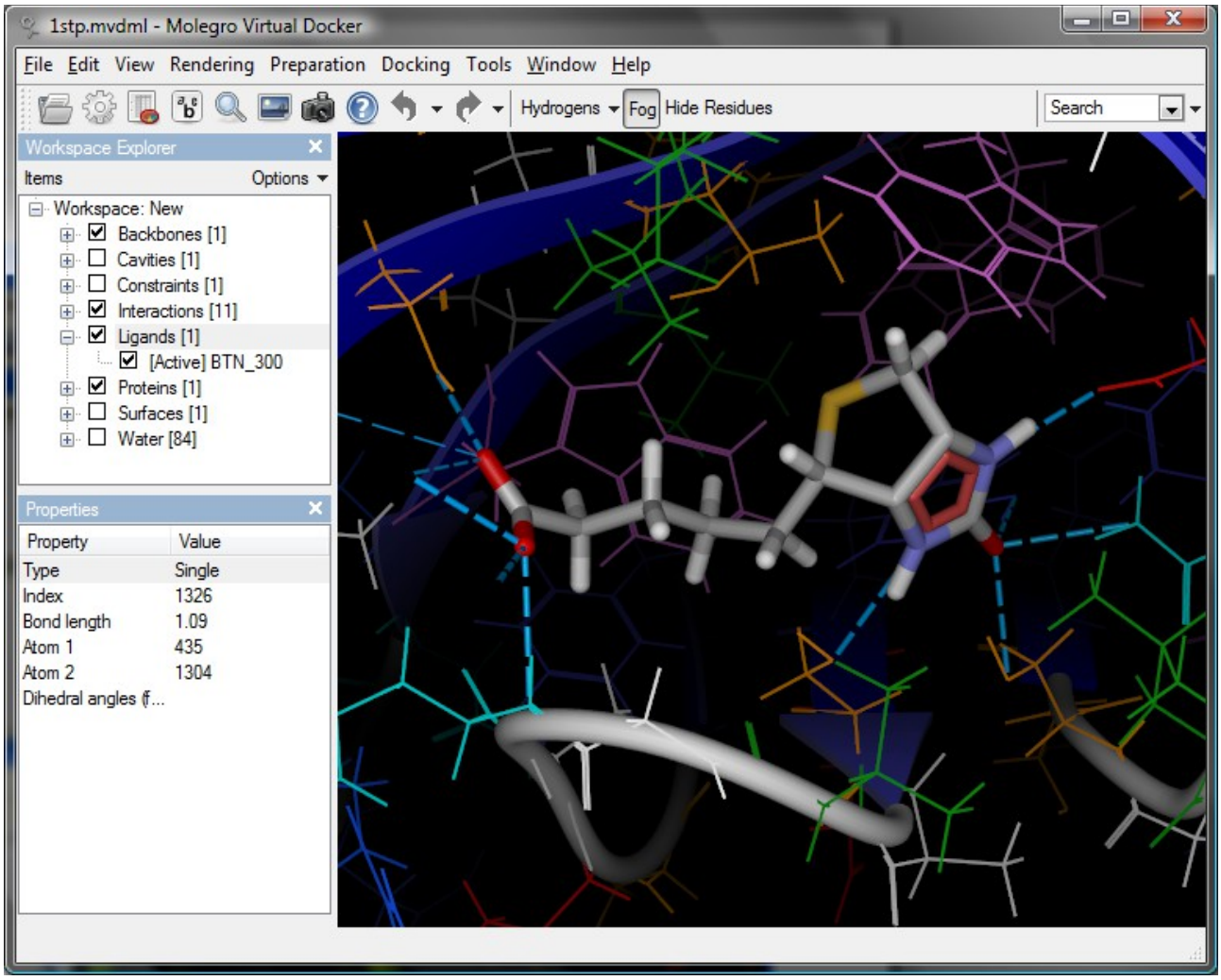


Figure 21: Main window showing different visualization styles.

The easiest way to get acquainted with the different drawing modes is to try the preset modes listed in the **Rendering** menu or to use the **Visualization Settings** dialog to inspect and modify visualization settings (described in Section 3.22). Afterwards, use the **Macro and Menu Editor** (described in Section 3.27) to explore which console commands that are used for a particular view.

Navigating the 3D World

Mouse actions available in the 3D world:

Function	Action
Zoom	By pressing both mouse buttons and moving up and down. By using scroll wheel. By using shift and left mouse button.
Free Rotation	Dragging mouse cursor while holding left mouse button down.
Drag Atom Rotation	While holding mouse over an atom: Dragging mouse (left mouse button down) will force the atom to follow the mouse cursor.
Free Translation	Dragging mouse cursor while holding right mouse button down.
Show Context Menu	Click and release right mouse button.

All rotations are centered about the rotational center.

This center can be chosen by invoking the context menu on an atom (right mouse button click) and selecting **Set as Rotational Center**. Another option is to choose **Fit to Screen** from the **Workspace Explorer** context menu. **Fit to Screen** will set the rotational center to the center of the bounding box enclosing the chosen molecule. If **Fit to Screen** is invoked from the **MVD Toolbar** or from the **Visualization Window** context menu, the new rotational center will be the center of the bounding box enclosing all visible molecules in the **Visualization Window**.

Manipulating Visualization Objects

All objects in the 3D world have context menu actions. These can be used for changing their properties, e.g. setting hybridization, partial charge, implicit hydrogens, or hydrogen bond types for atoms and bond order or bond flexibility for bonds. See Section 4.3 for more details.

3.6 Console Window

The **Console Window** (at the bottom of the screen) displays information, warnings and errors. The input field at the bottom of the console window

allows the user to enter console commands. The amount of information in the console can be controlled with the associated context menu (right mouse button click) - e.g. info, warnings, and debug messages can be turned off.

3.7 Clipping Planes

Clipping Planes allows you to change the clipping planes of the visualization window, i.e. how close and how far away objects are drawn. This can for example be useful if you want to visualize the interior of a protein or a ligand deeply buried inside a macromolecule.

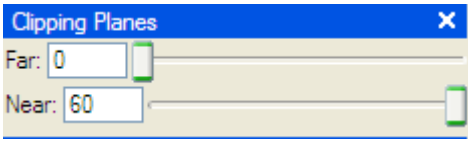


Figure 22: Clipping Planes dockable window.

Clipping Planes can be enabled by choosing **Window** | **Clipping Planes...** from the menu bar. Clipping Planes are enabled when the **Clipping Planes** window is shown and disabled when it is closed. Adjust the near and far slider until the desired region is shown.

3.8 Creating a Search Space

In MVD, a **Search Space** is defined by a position (x,y,z) and a radius. The search space is mainly used to define the volume a docking simulation explores, but may be used for other purposes as well: for instance cropping molecules in the workspace, making a partial molecular surface, or hiding molecules outside the search space.

The **Search Space** button on the main GUI toolbar makes it easy to toggle the search space on and off.

Toggling the **Search Space** button is identical to using the checkbox for the search space item in the Workspace's **Constraints** category, with one exception: if there is no search space in the workspace, pressing the **Search Space** button will invoke the **Search Space Setup** dialog:

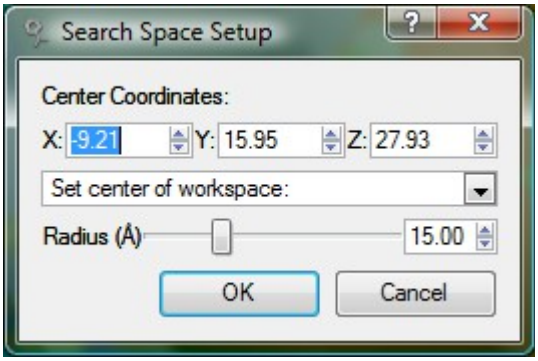


Figure 23: Search Space Setup Dialog.

The **Search Space Dialog** makes it possible to set the center and the radius of the search space. It is possible to directly set the center coordinates of the search space, or to place the center on either:

- The center of the proteins in the workspace
- The center of a ligand or a pose in the workspace (only the first 10 molecules in each category are shown)
- The center of a cavity (only the 10 largest cavities are shown)
- The center of all selected objects

The Search Space Setup dialog may also be invoked from the application menu using **Preparation** | **Search Space Setup**, by using the context menu on the the search space item in the Workspace Explorer's **Constraints** category, or by using the context menu on any atom or selection in the 3D view and selecting **Set as center of search space**.

3.9 Hiding Distant Residues

The **Hide Residues** dialog can be invoked by pressing the **Hide Residues** button in the **MVD Toolbar**. In order to show all protein residues again, select the **Hide Residues** button on the **MVD Toolbar**.

The **Hide Residues** dialog (see Figure 24) allows you to hide non-relevant residues and molecules. It is also possible to only display specific residue types.

It is possible to hide objects based on their distance to one of the following objects: any *ligand* or *pose* in the workspace (only the first twenty are listed), any *cavity*, any *selected objects*, a *search space*, or *marked residues* (when using the Protein Preparation dialog).

If a ligand is chosen, the minimum distance between all atoms in the ligand and all atoms in a given residue is calculated. This residue is then hidden if it is farther away than the chosen proximity distance. Poses work the same way.

For cavities the distance to each single cavity point is considered when hiding objects. The residues and molecules are dynamically shown/hidden when the **Proximity** slider is moved.

The lower pane of the **Hide Residues** dialog allows you to restrict the types of residues shown by toggling the appropriate button. If a given residue type is not within proximity distance as defined in the panel above, the button corresponding to the type will be grayed and can not be toggled.

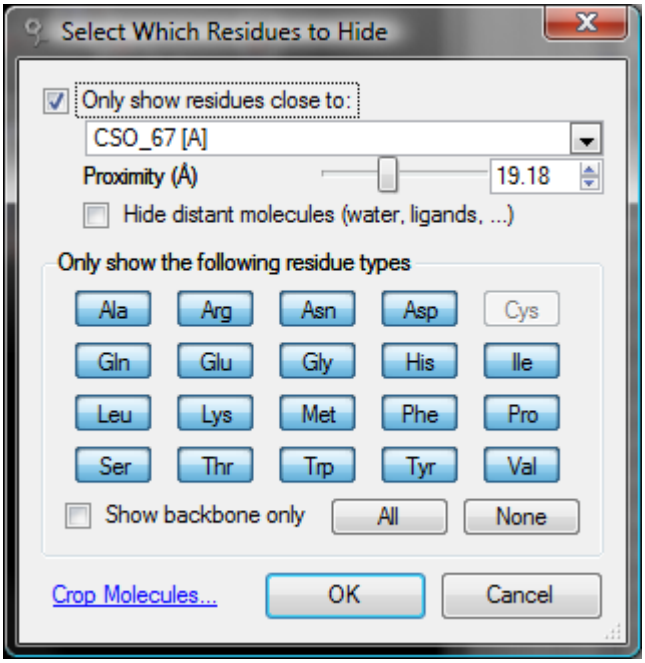


Figure 24: Hide Residues dialog.

The **Show backbone only** check box can be used to toggle whether side-chains are visible or not.

Cropping. It is possible to delete molecules from the workspace in order to remove non-relevant regions. To crop molecules, invoke the **Hide Residues** dialog and adjust the settings until the desired residues and molecules are displayed before clicking the **Crop Molecules...** button. A dialog will show which structures will be kept (the checked molecules) and which will be discarded.

Notice that proteins are split and cropped on a per-residue basis: hidden residues will be discarded and visible residues will be kept. All other molecule types are kept or discarded in their entirety.

3.10 Workspace Finder

The **Workspace Finder** located in the far right side of the **MVD Toolbar** (see Figure 25) allows you to quickly search for molecule names and residue/atom IDs in the workspace. The **Workspace Finder** is invoked by typing characters

in the search box (text field). A result is selected by pressing the **Return** key. Pressing the **Escape** (Esc) key or mouse-clicking outside the **Workspace Finder** window will cancel the current search query.

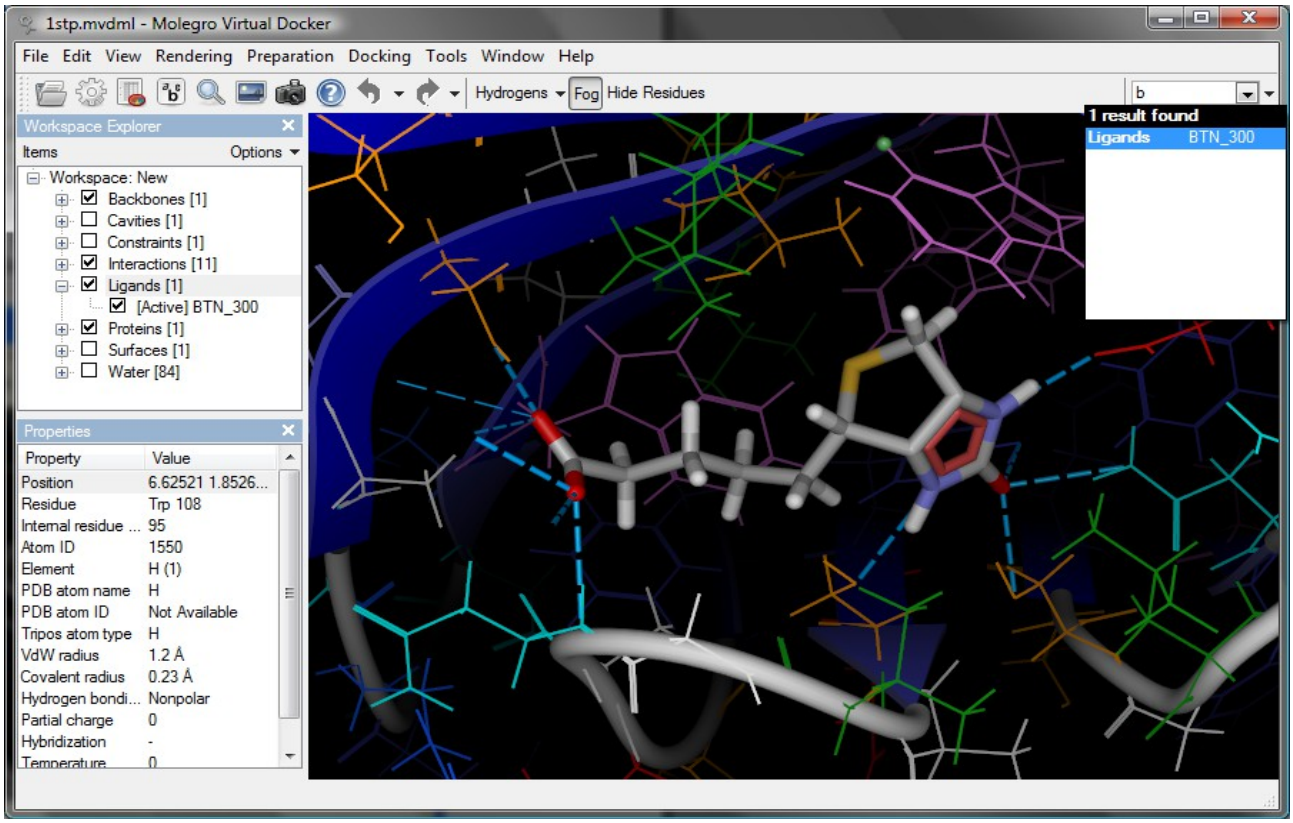


Figure 25: Workspace Finder dialog.

When a name or ID number (or part of it) is typed in the search box, the **Workspace Finder** will present a list of matches (a maximum of 30 matches is returned). It is also possible to search in atom coordinates by prepending the search with a '!' (e.g. searching for '!1.23' will return atoms where one of the coordinates starts with 1.23).

By default, the **Fit to screen** option is enabled so that items (molecules, residues, or atoms) are fitted to the **Visualization Window** while browsing the list of results found. The **Fit to screen** option can be disabled in the options panel invoked by pressing the small button on the right hand side of the **Workspace Finder** search box.

3.11 Sequence Viewer

The **Sequence Viewer** dialog (see Figure 26) allows you to inspect protein residues in an easy manner. The dialog can be invoked by selecting **Window** | **Sequence Viewer**' or using the **Ctrl-Shift-S** keyboard shortcut.

Using the context menu on the **Sequence Viewer** window it is possible to

select residue atoms in the **Visualization Window**, hide non-selected residues, change between one and three-letter residue names, and toggle details about secondary structure. Residues near cavities are indicated with a green ribbon (the distance threshold may be set using the sequence viewer's context menu) and broken protein chains are indicated with vertical lines between residue endpoints.

Detailed information about residue name, index, and secondary structure assignment is available in the tool tip, which can be invoked by focusing the mouse on a specific residue in the Sequence Viewer.

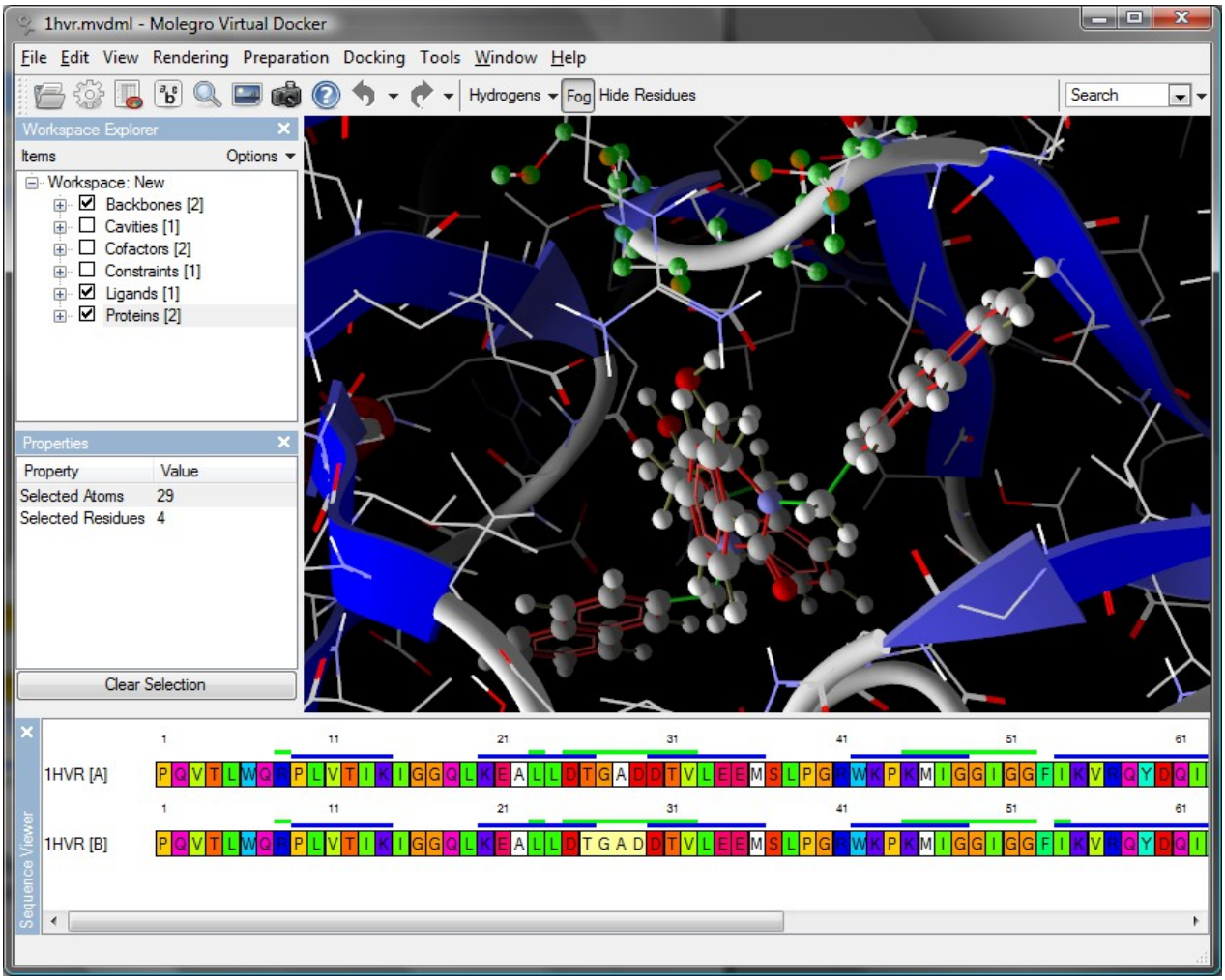


Figure 26: Sequence viewer with selection of four residues highlighted in the Visualization Window.

3.12 Workspace Properties

Workspaces can contain user-specified notes. Further, the title of the workspace can be changed using the **Workspace Properties** dialog. The **Workspace Properties** dialog can be found in the **Edit Properties...** context menu on the **Workspace** item in the **Workspace Explorer** or via **Edit** | **Workspace Properties...** (see Figure 27).

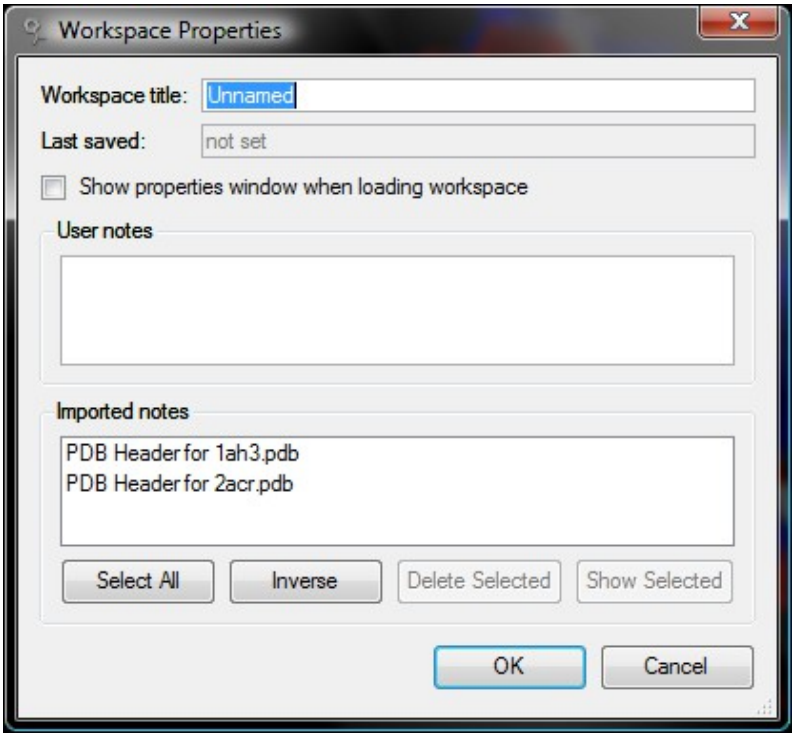


Figure 27: Workspace Properties dialog.

3.13 Measurements and Annotations

Distances and angles can be measured directly in the 3D world (see Figure 28).

If two atoms are selected, the distance between them will be shown in the **Properties Window**.

If three connected atoms are selected, the angle that they span will be shown in the **Properties Window**.

If no atoms are selected, and a bond is highlighted, the field **Torsion Angles** in the **Properties Window** will show the torsion angle(s), defined through this bond.

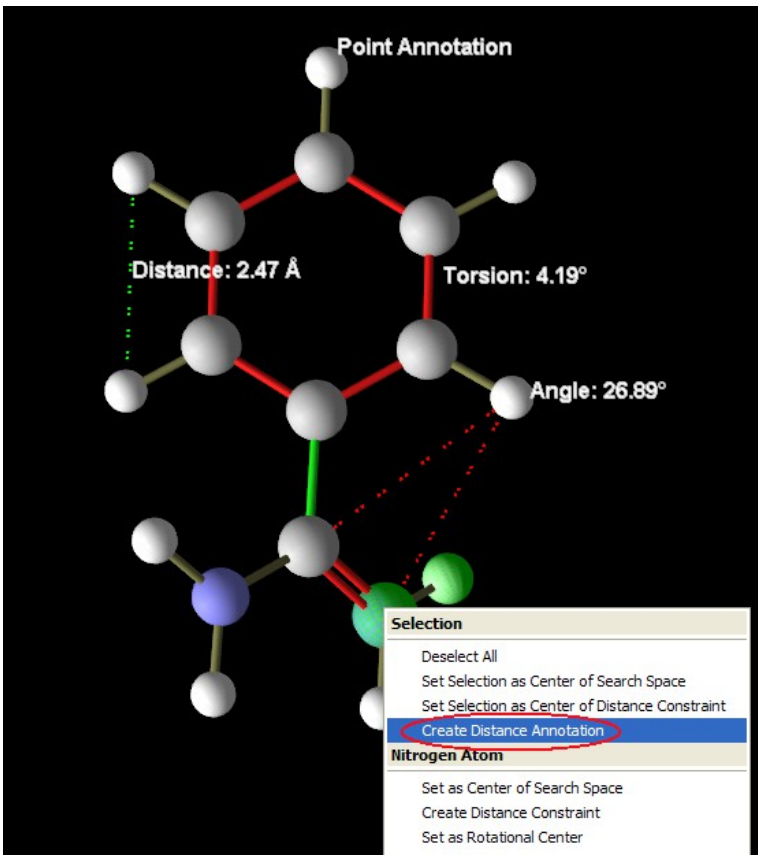


Figure 28: Annotations and measurements.

Measurements can also be made permanent as annotations. There are different kinds of annotations. To create annotations, select 1-4 atoms and use the context menu (right-click mouse button) and choose **Create ... Annotation**. The text can be edited before the annotation label is created. Annotations are added to the **Workspace Explorer** category: **Annotations**. Annotations can also be removed from the workspace using the context menu available from the Annotations category in the Workspace Explorer window.

3.14 Selection of Atoms, Amino Acids, Rings, and Molecules

Atoms can be manually selected in the Visualization Window using the mouse.

Using the context menu when focusing on a specific atom it is also possible to select/deselect atoms, molecules, molecules (carbon only), rings (for ligands/cofactors/poses), and amino acids (for proteins).

3.15 Custom Coloring of Atoms, Amino Acids, and Molecules

The atoms in a selection can be set to a custom color using the context menu (invoked by pressing the right-mouse button on a given atom).

Entire molecules can be set to a custom color using the Workspace Explorer

context menu by selecting either **Set Custom Color...** or **Set Custom Color (Carbons Only)...**.

Custom Coloring is persistent - it will persist after changing rendering/coloring styles, and takes precedence over any coloring style.

The Custom Coloring can be cleared using the **Clear Custom Coloring** option from the Workspace Explorer context menu or from the Visualization Window context menu (when focusing on a given atom).

Notice that aromatic ring indicators (pseudo-bonds) and single-colored bonds will only have custom coloring applied, if the entire molecule is selected (or if the **Set Custom Color** command is invoked from the Workspace Explorer context menu).

The Custom Coloring information is stored together with the atoms in MVDML files and will be used every time the MVDML workspace file is opened in MVD.

3.16 Creating Labels

To create labels use the **Create Label** dialog, which can be invoked via **Create Labels...** in the **Workspace Explorer** context menu (on molecular categories: **Proteins**, **Ligands**, and **Poses**) or via the **Tools** | **Labels** menus.

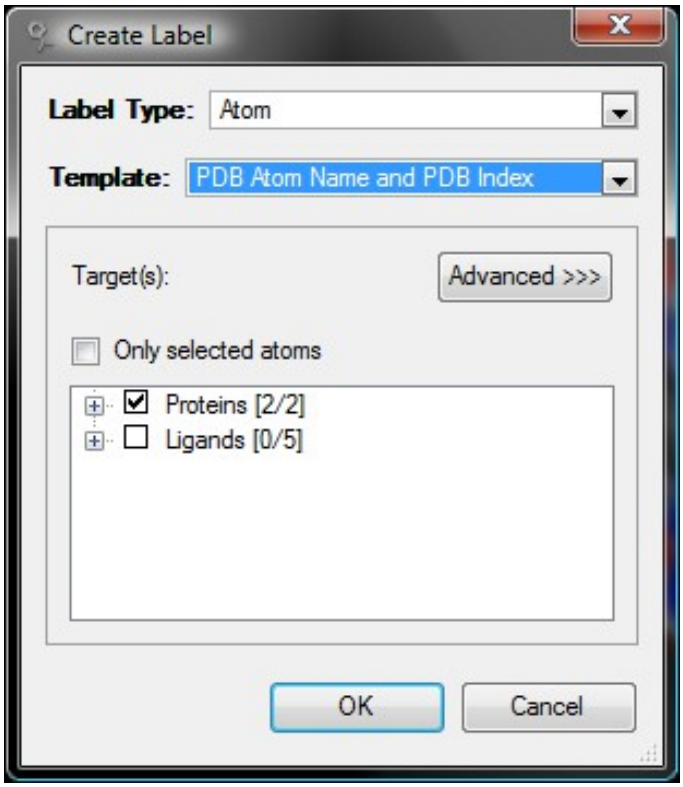


Figure 29:Creating a new label.

The **Create Label** dialog makes it possible to label different *object levels*: atoms, bonds, molecules, or residues. The labels can be chosen from a list of

standard templates or constructed from a list of available variables (using the **Advanced** tab).

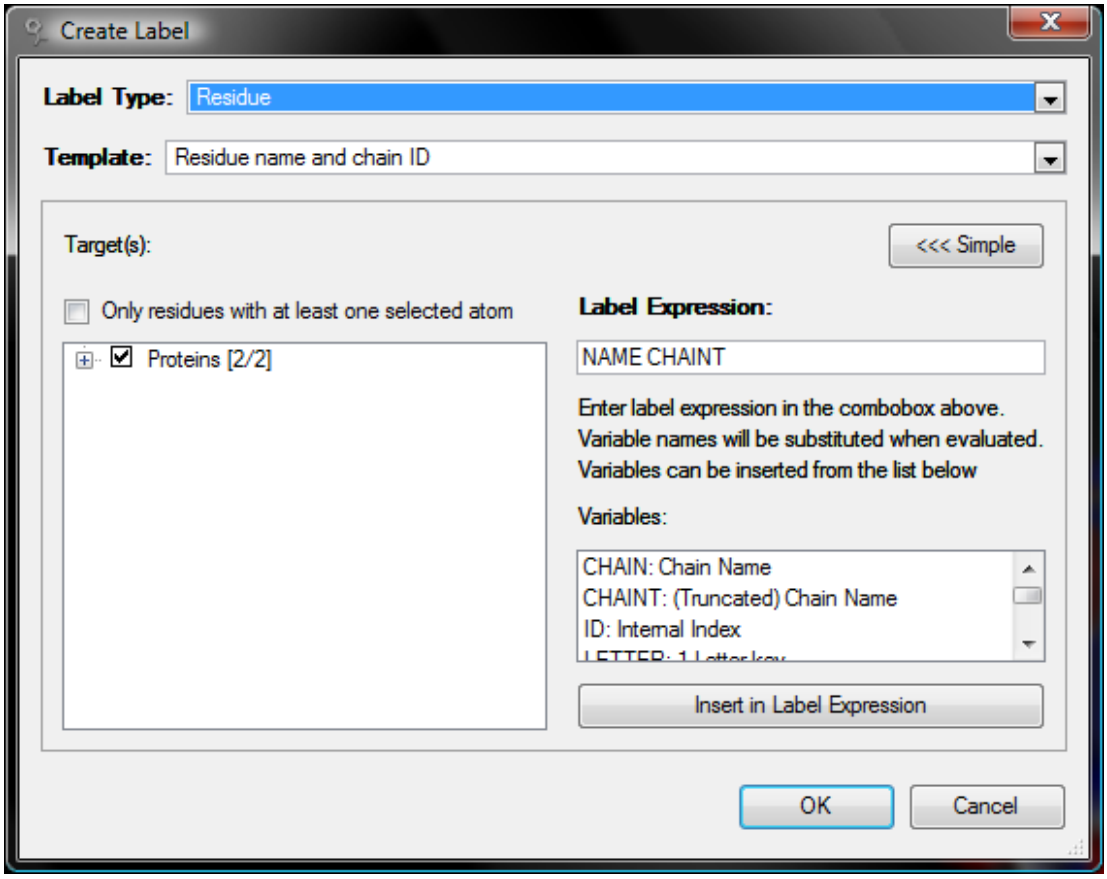


Figure 30:Advanced label expression dialog.

Labels will occur in the **Labels** category in the **Workspace Explorer** - assigned in groups (one group for each molecule). Labels can be removed or hidden using the context menu or by pressing the labels tool bar button.

3.17 Creating Molecular Surfaces

Surfaces can be created for all molecular objects via **Create Surface...** from the context menu in the **Workspace Explorer** or via **Tools** | **Surfaces**.

In MVD surfaces are created by probing points on a uniformly spaced grid. It is possible to adjust the grid resolution (**Resolution**) and probe size (**Probe Radius**) under **Advanced** settings.

Two types of surfaces are available:

Expanded Van der Waals – this is an approximation to the surface created by expanding the Van der Waals radius of each atom with the **Probe Radius**.

Molecular surface – this is an approximation to the surface defined by the contact area of the probe and Van der Waals sized spheres.

It is also possible to restrict the surface to a volume defined by the current search space by enabling **Restrict to 'search space'**.

Surfaces can be colored by **Hydrophobicity**, **Electrostatic Potential**, or **Solid Color**. Surfaces can be drawn transparently, as dots, lines, or solid polygons.

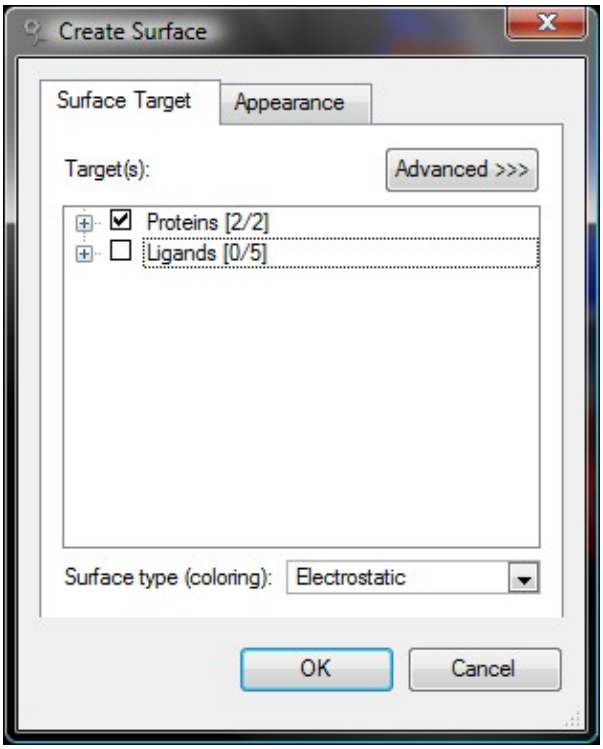


Figure 31: Creating a new surface.

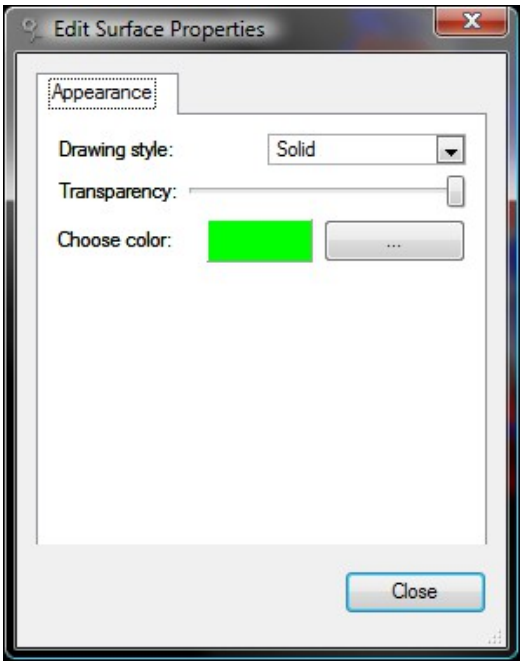


Figure 32: Changing surface appearance.

3.18 Creating Protein Backbone Visualizations

The backbone of the protein can be visualized by using the **Create Backbone Visualization** dialog. The dialog can be invoked by using the context menu on the **Proteins** category (or a single protein item) in the **Workspace Explorer**.

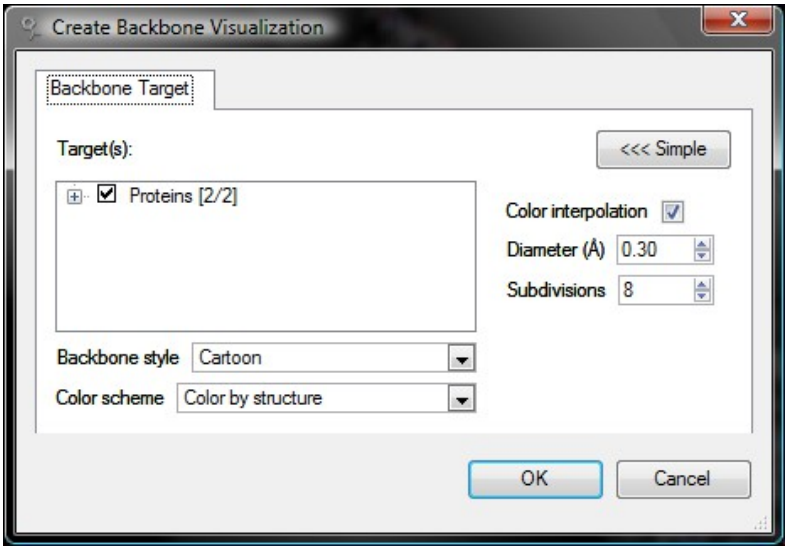


Figure 33: Creating a new backbone.

The **Create Backbone Visualization** dialog allows you to select which proteins (or protein chains) the backbone should be visualized for.

Three main graphics styles can be used. The **Cartoon** style visualizes the

secondary structure of the protein(s) using arrows to represent beta sheets and helical lines for alpha helices (see Figure 34).

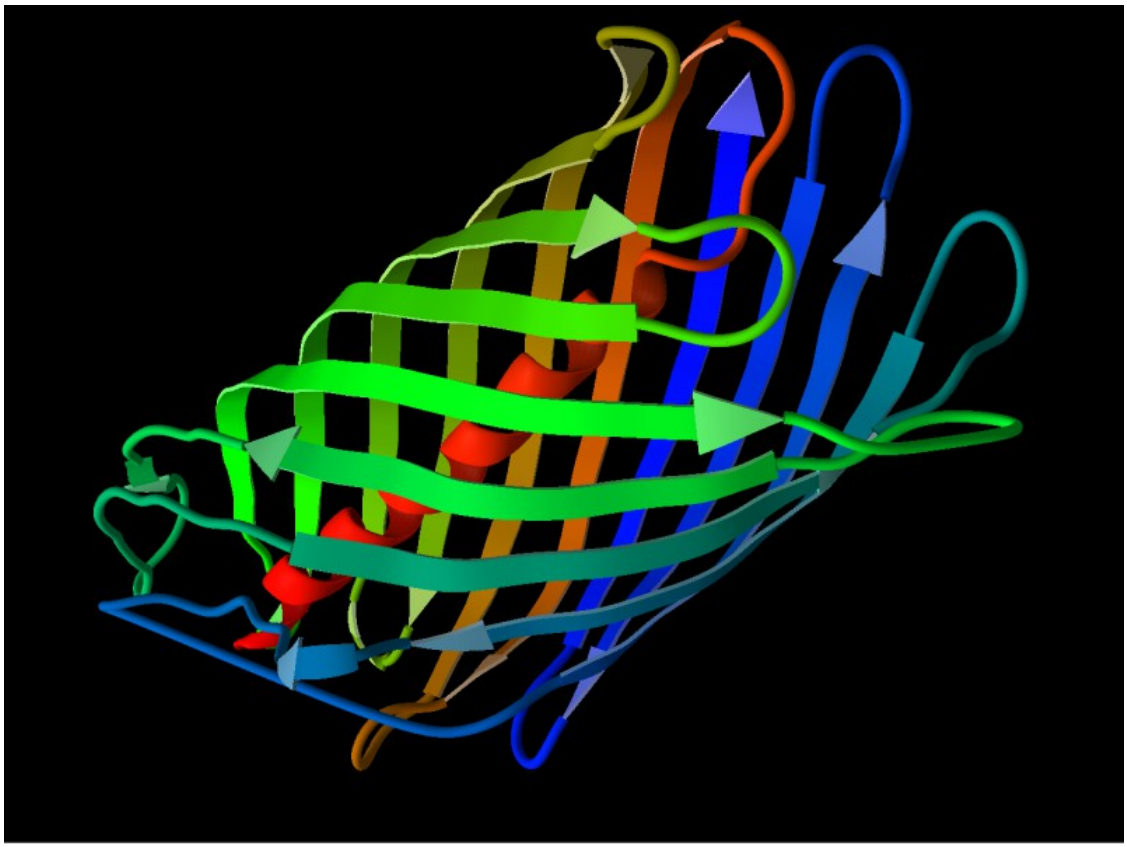


Figure 34: Cartoon graphics style.

If the **Tube** graphics style is used, the backbone is visualized as a spline (a piecewise parametric polynomial curve) interpolating the positions of the alpha carbons in the backbone (see Figure 35).

Figure 35: An example of a protein backbone using the Tube graphics style.

The **Difference Tube** graphics style requires two superimposed protein chains to be present in the workspace. The radius of the 'difference' tube will be proportional to the distance between a C-alpha atom from the selected protein chain and the nearest C-alpha atom of any other chain in the workspace (i.e. the C-alpha atoms compared are not based on a sequence alignment). This makes it possible to visualize where two superimposed structures differs the most.

It is also possible to set the color scheme for the backbone. **Color by structure** colors the backbone based on the secondary structure information (alpha helices are colored yellow, beta sheets are colored blue, and coil is colored gray). **Color by residue position** colors the backbone based on the residues order of occurrence creating a rainbow color effect. **Color by chain** colors each individual protein chain in a different color. **Color by atom** colors the backbone by using the currently shown color of the protein backbone atoms (the color used is taken from the C-alpha atom).

On the advanced panel, the **Color interpolation** check box allows you to determine whether the backbone color should be interpolated between the atoms it passes through or should be held constant between atoms. **Diameter (Å)** sets the width of the backbone in angstrom, **Subdivision** sets the resolution of the backbone (the number of subdivisions between each residue in the protein).

Backbones appear in the **Backbones** category in the **Workspace Explorer** and can be removed via the context menu or hidden using the check box.

3.19 Making Screenshots

Screenshots can be made by choosing **Window** | **Capture Screen**.

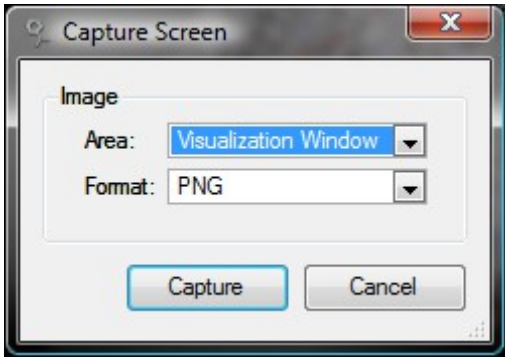


Figure 36: Screen Capture dialog.

It is possible to specify whether to capture the **Visualization Window** only (the 3D view) or the entire **Desktop** (see Figure 36). The captured region can be saved in JPG, BMP, or PNG file formats.

3.20 Sidechain Minimization

MVD allows you to minimize a protein with respect to itself and other structures in the workspace. The minimization is performed using a fairly simple forcefield (it uses the PLP-potentials for steric and hydrogen bonding interactions, and the Coulomb potential for the electrostatic forces as defined in Appendix I: MolDock Scoring Function). Only torsion angles in the sidechains are modified during the minimization – all other properties (including bond lengths and backbone atom positions) are held fixed.

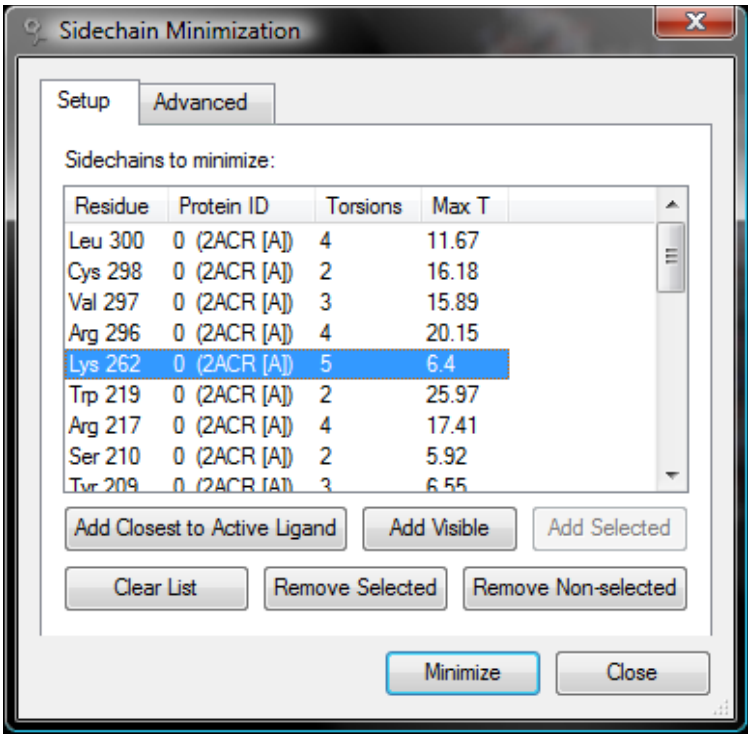


Figure 37: The Sidechain Minimization dialog.

The **Sidechain Minimization** dialog can be invoked from **Tools** | **Sidechain Minimization** (see Figure 37).

The **Setup** tab on the dialog controls which sidechains to minimize. Several options exist for choosing the sidechains:

Add Closest to Active Ligand - This will choose all sidechains which are close enough to the active ligand to interact with it. More precisely: for each given sidechain, a sphere bounding all possible configurations of the sidechain is calculated, and it is tested whether any atom in the active ligand is close enough to make a steric contact with an atom in this bounding sphere (for the 'MolDock' potential, all steric contacts are cut off at a distance of 6.0 Å). Notice that the 'Active ligand' can be set in the Workspace Explorer window: it is the ligand which name is prepended with an '[Active]' label.

Add Visible - This will add all sidechains which are currently visible in the 3D Visualization window. This feature can be used together with the **Hide Residues** dialog where it is possible to hide sidechains depending on the distance from some given object.

Add Selected - This feature allows for selecting sidechains directly in the 3D Visualization window. A sidechain is selected if one or more atoms inside it are chosen.

Clear List - Removes all sidechains from the list.

Remove Selected - Removes all sidechains that are currently highlighted in the sidechain list view.

Remove Non-selected - Removes all sidechains that are not highlighted in the sidechain list view.

The following columns display information about the selected sidechains:

Residue - The residue name/id.

Protein ID - The protein (or protein chain) ID and name.

Torsions - The number of degrees of freedom in the given sidechain. The degrees of freedom that are minimized during the docking simulation are the torsional angles in the sidechain.

Max T - The temperature factor or B-factor is a measure of how much a given atom vibrates around its position in the crystallographic model. This can be useful since a high B-factor may indicate that the residue is likely to be flexible. **Max T** is the single highest temperature factor of all (heavy) atoms in the sidechain.

Columns in the list can be toggled on or off using the context menu on the sidechain list view.

The Advanced Tab

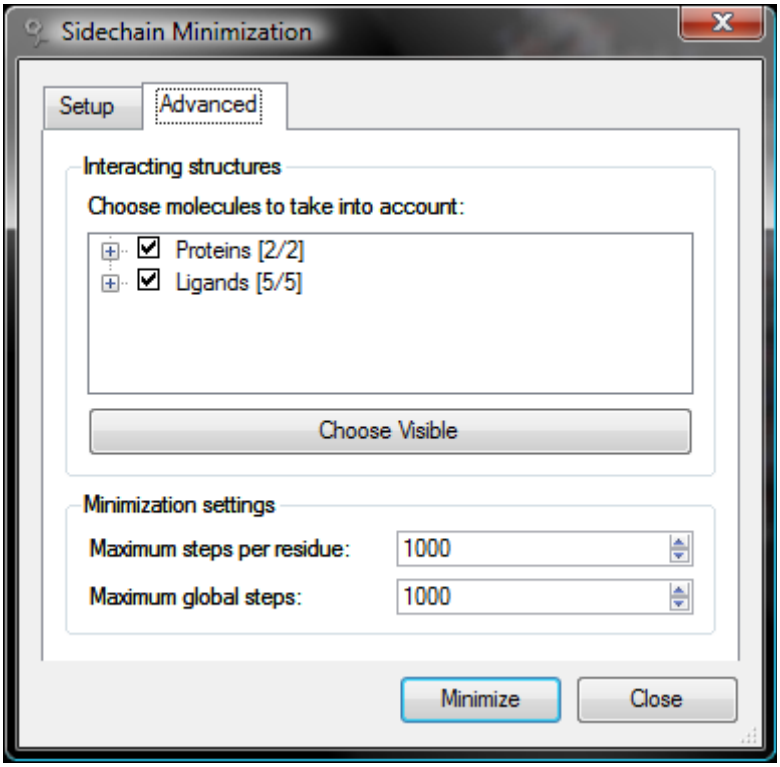


Figure 38: The Sidechain Minimization 'Advanced' tab.

The **Advanced** tab allows you to determine which structures in the workspace should interact with the sidechains during the minimization. By default all

structures (ligands, cofactors, water, ...) are taken into account when minimizing. If you work simultaneously with multiple conformations of the same receptor in the workspace, make sure that only the specific conformation to be minimized is selected as an interacting structure.

The **Choose Visible** button selects all structures which are currently visible in the workspace as interacting structures.

The lower panel (**Minimization settings**) handles the setup of the minimizations algorithm.

The minimization algorithm is Nelder-Mead simplex minimization, first running a specified number of steps independently for each residue (**Maximum steps per residue**) and afterwards performing a global minimization run on all residues simultaneously (**Maximum global steps**).

The minimization procedure is started by pressing the **Minimize** button. After the minimization procedure completes a new *receptor conformation* will be added to the workspace (usually it takes just a few seconds, but this depends on the number of residues being minimized) .

Also notice that new columns are added to the sidechain list after the minimization has completed: **E_before**, which is the energy before the sidechain has been minimized, **E_after** which is the energy after and **dE** which is the difference in energy. Notice that these energies are not measured in chemically relevant units, and that their magnitude will depend on which structures were taken into account during the minimization.

3.21 Working With Multiple Receptor Conformations.

When docking with sidechain flexibility or after sidechain minimizations have been conducted, new *receptor conformations* are added to the workspace.

A *receptor conformation* is a list of torsional changes to an existing receptor (which can be one or more proteins chains – each protein chain will have its own entry under **Proteins** in the **Workspace Explorer**). Notice that a receptor conformation is not an isolated entity – it always exists in the context of one or more proteins or protein chains.

When new receptor conformations are added to the workspace they will appear in a drop-down box in the **Workspace Explorer** window (Figure 39):

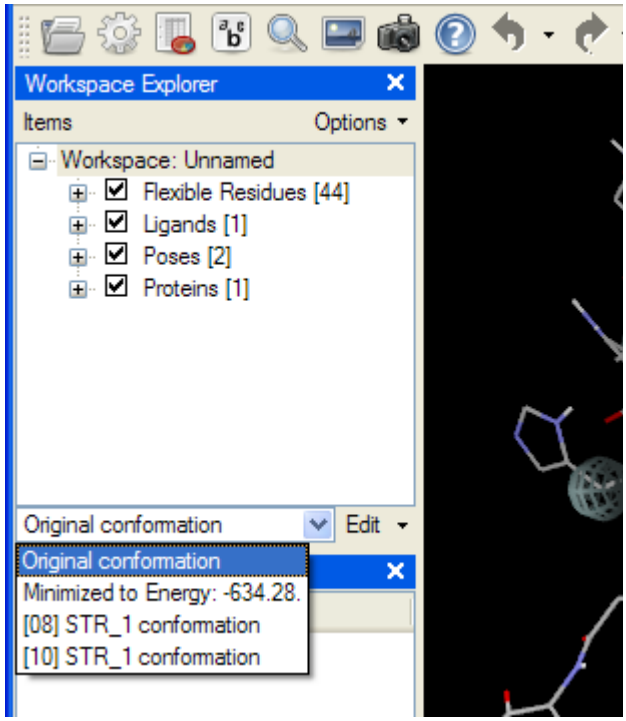


Figure 39: Receptor conformation list.

In order to manipulate receptor conformations, select the appropriate conformation and press the **Edit** drop-down button in the lower-right corner of the **Workspace Explorer** (Figure 40).

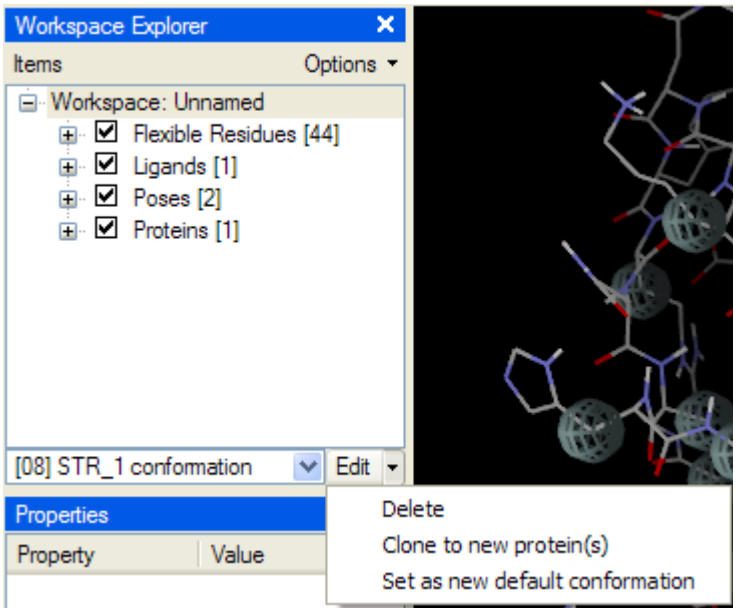


Figure 40: Actions for receptor conformations.

You can delete a conformation (**Delete**) or choose **Clone to new protein(s)** to clone the current conformation to one or more new proteins (if the conformation consists of torsional changes to more than one protein (or

protein chain), a clone will be made for each protein). The last option (**Set as new default conformation**) will make the currently selected conformation the only conformation in the workspace – all other conformations will be discarded, and the original conformation will no longer be available.

Conformations are saved together with the workspace in the MVDML file format. Notice that before saving, the program will always change to *default conformation* and when loading workspaces the *default conformation* will always be the currently selected conformation at startup.

3.22 Visualization Settings Dialog

The graphical settings for the 3D visualization can be adjusted by selecting **Rendering** | **Visualization Settings Dialog**.

Graphical Styles and Coloring Schemes

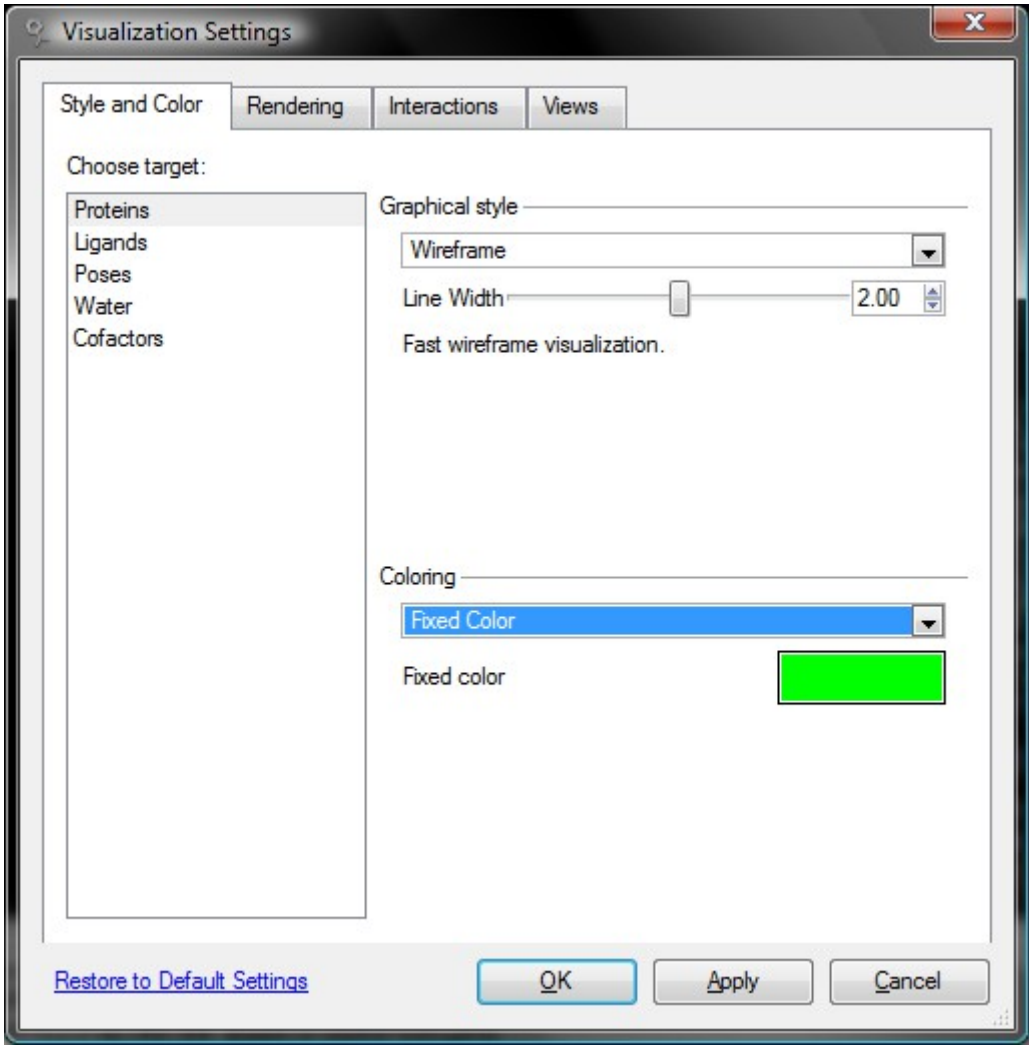


Figure 41:The Visualization Settings dialog.

From the **Style and Color** tab, select a category from the list on the left side of the tab (one of 'Proteins', 'Ligands','Poses','Water', and 'Cofactors') and adjust either its graphical style or color scheme.

The following graphical styles can be chosen:

- **Ball and Stick**. Atoms are drawn as spheres (balls), and bonds are drawn as cylinders (sticks). The **Atom Scale** parameter sets the fraction of the Van der Waals radius that is used as radius for the sphere. **Bond Scale** is the diameter of the bonds in Ångstrom. This is the preferred graphical style for modifying and inspecting bond and atom properties (since the bond order is visualized and the atoms are easy to select).
- **Stick**. Bonds are drawn as cylinders. **Bond Scale** is the diameter of the bonds in Ångstrom.
- **Spacefill (CPK)**. Atoms are drawn as spheres (balls). Bonds are not drawn. The **Atom Scale** parameter sets the fraction of the Van der Waals radius that is used as radius for the sphere.
- **Wireframe**. This is by far the fastest way to draw molecules. Bonds are drawn as lines between atoms. No atoms are drawn (but notice that it is still possible to do atom selections in the GUI). Notice all bonds are drawn as single lines (double bonds and delocalized bonds are also drawn as single lines). It is possible to adjust the line width in pixels (Notice that not all OpenGL implementations support non-integer line widths).

The following coloring styles can be applied to all molecules:

- **Fixed Color** - A user-defined color.
- **Color By Element (CPK)** - Atoms are colored according to element type.
- **Color By Id (or Chain)** - Molecules are colored according to their internal molecule ID (i.e. a single ligand will be uniformly colored, but all ligands will have different colors).
- **Color By Id (carbons only)** - Same as above, except only carbons are colored using this scheme. Other atoms are colored according to element type.
- **Color By Hydrogen Bond Type** - Colors atoms according to hydrogen bonding properties (donors are red, acceptors green and atoms capable of both donating and accepting hydrogens are yellow).
- **Color By Partial Charge** - Colors according to electrostatic partial charge (blue corresponds to positive charge, red to negative charge).

The following can only be applied to proteins:

- **Color By Temperature (B-Factor)** - The temperature factor is a

measure of how much a given atom vibrates around its position in the crystal structure. Notice that this information is not always present in PDB-files, and that it is sometimes used for other purposes. The colors will be interpolated between blue for the minimum temperature and red for the maximum temperature.

- **Color By Amino Acid Type** - Colors proteins according to their residue type.
- **Color By Shapely Residue Scheme** - Same as above with alternative colors.
- **Color By Residue ID** - Colors according to residue ID (rainbow effect).
- **Color By Secondary Structure** - Colors according to secondary structure (red for helices, blue for strands and yellow for turns).
- **Color By Hydrophobicity** – Residue atoms are colored according to the hydropathy index proposed by Kyle and Doolittle in 1982 (see http://en.wikipedia.org/wiki/Hydropathy_index for details). Hydrophilic residues are colored red, hydrophobic residues are colored blue.

Rendering Settings

The **Rendering** tab (Figure 42) on the **Visualization Settings** dialog allows you to customize the rendering behavior.

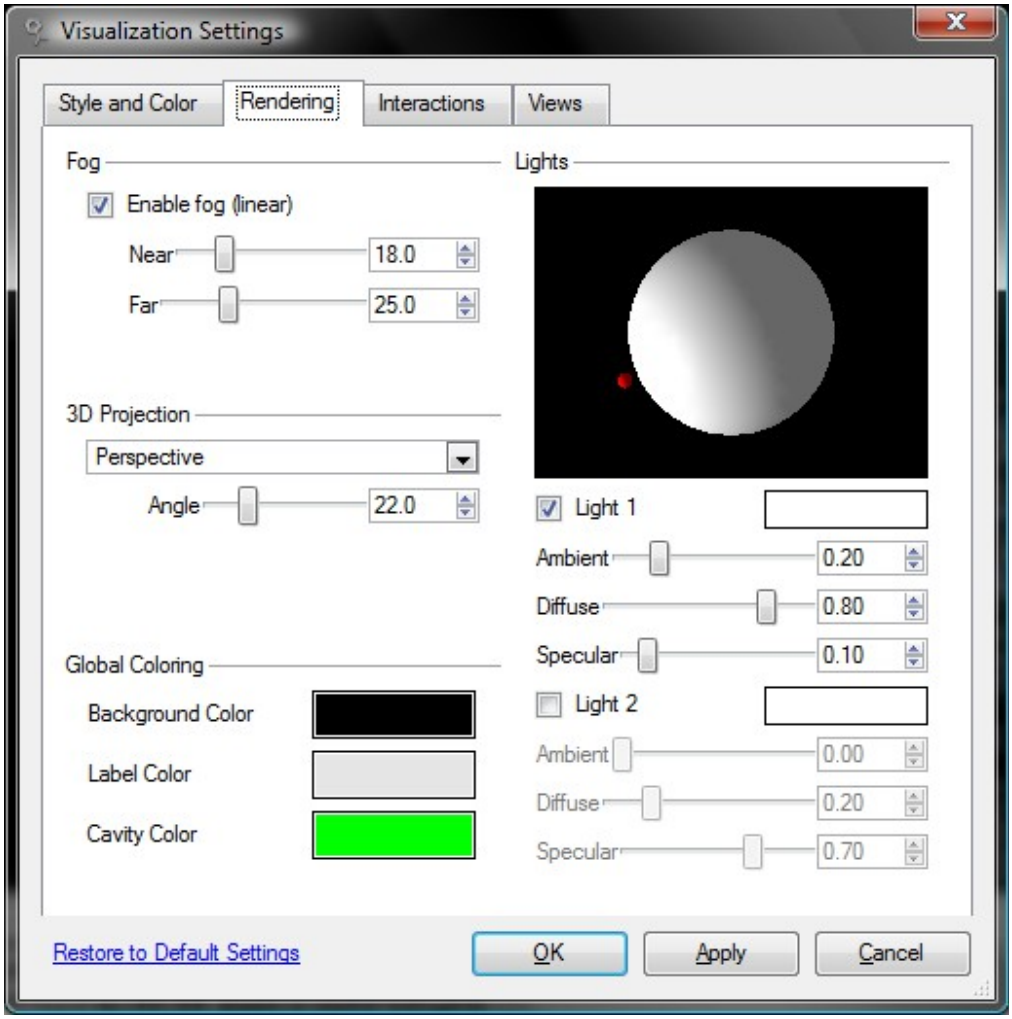


Figure 42:The Visualization Settings Rendering options.

The **Fog** settings enables or disables fog. It is possible to adjust when the fog should begin (the **Near** value) and when the fog should reach its maximum density (the **Far** value).

The **3D Projection** settings manage the perspective projection. In **Perspective** projection objects farther away from the viewer appear smaller (the magnitude of this effect can be controlled by adjusting the field-of-view **Angle** parameter). In **Orthographic** projection object sizes are independent of their distance from the viewer.

The **Global Coloring** settings allow you to adjust the background color, the color labels are drawn with, and the color cavities (predicted binding pockets) are drawn with.

The **Lights** section controls the global lightning of the 3D world. It is possible to enable one or two light sources. Their positions can be adjusted directly in the 3D sphere view. The light source color can be changed by clicking the color

selector next to the light checkbox.

OpenGL Lights contain three different parts: **Ambient** light always reaches an object, independent of its position relative to the light source. **Diffuse** lightning is dependent on whether the object faces the light source or faces away from it. The reflected light is emitted equally in all directions. **Specular** lightning is also dependent on the objects' orientation towards the light source, but the reflected light is emitted mainly in the direction of the reflected light ray (creating 'highlights').

Interactions

The **Interactions** tab (Figure 43) on the **Visualization Settings** dialog allows you to customize the appearance of hydrogen bonds (energy thresholds, thickness of bond, and color) and electrostatic interactions (energy thresholds and color) shown in the Visualization Window.

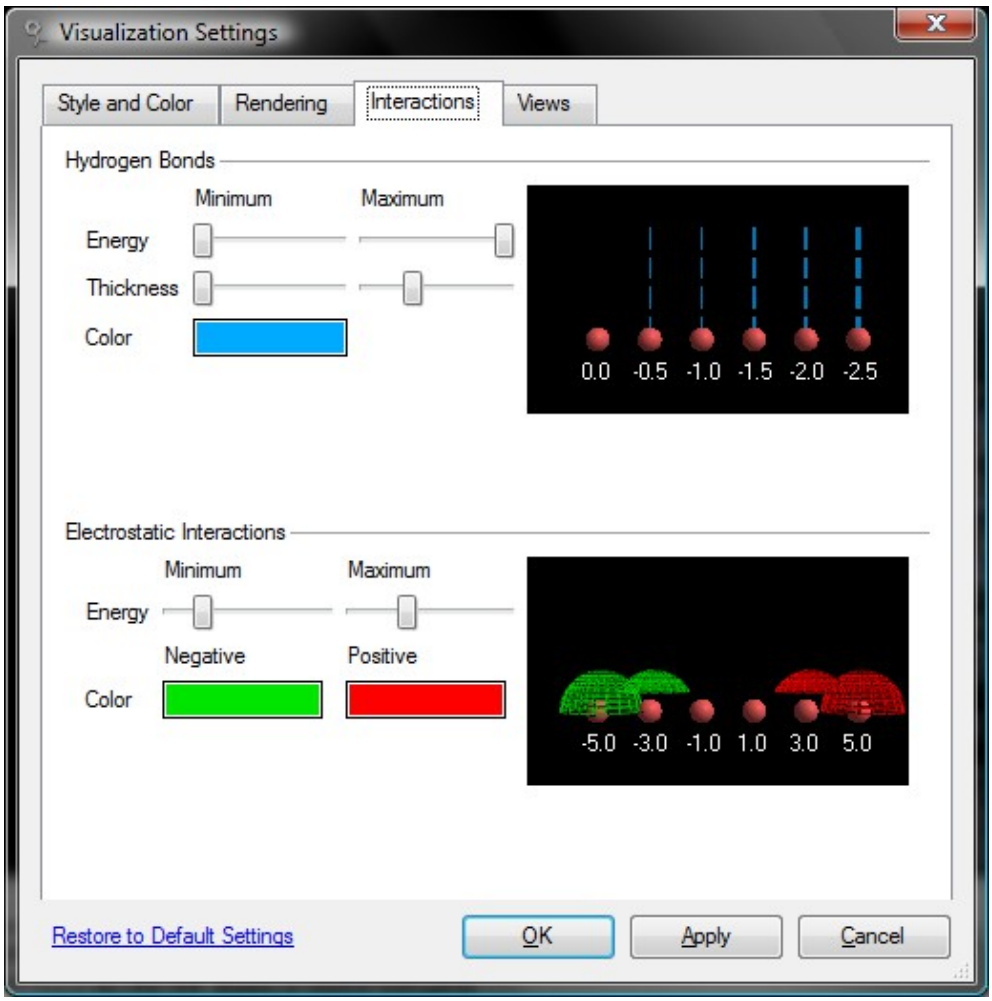


Figure 43:Settings for hydrogen bonds and electrostatic interactions.

Preset Views

The **Views** tab (Figure 44) in the **Visualization Settings** dialog controls the preset views (the macros residing under the **View** menu item on the main window menu bar).

The upper panel on the tab allows you to activate a preset view (by pressing the **Select** button') or delete a view (the **Delete** button). Notice that when deleting a view, you are not able to recover it unless you restore all macros (this can be done by choosing **Edit** | **Macro and Menu Editor** and pressing **Restore all macros**, but notice that all user changes to the macros will be lost).

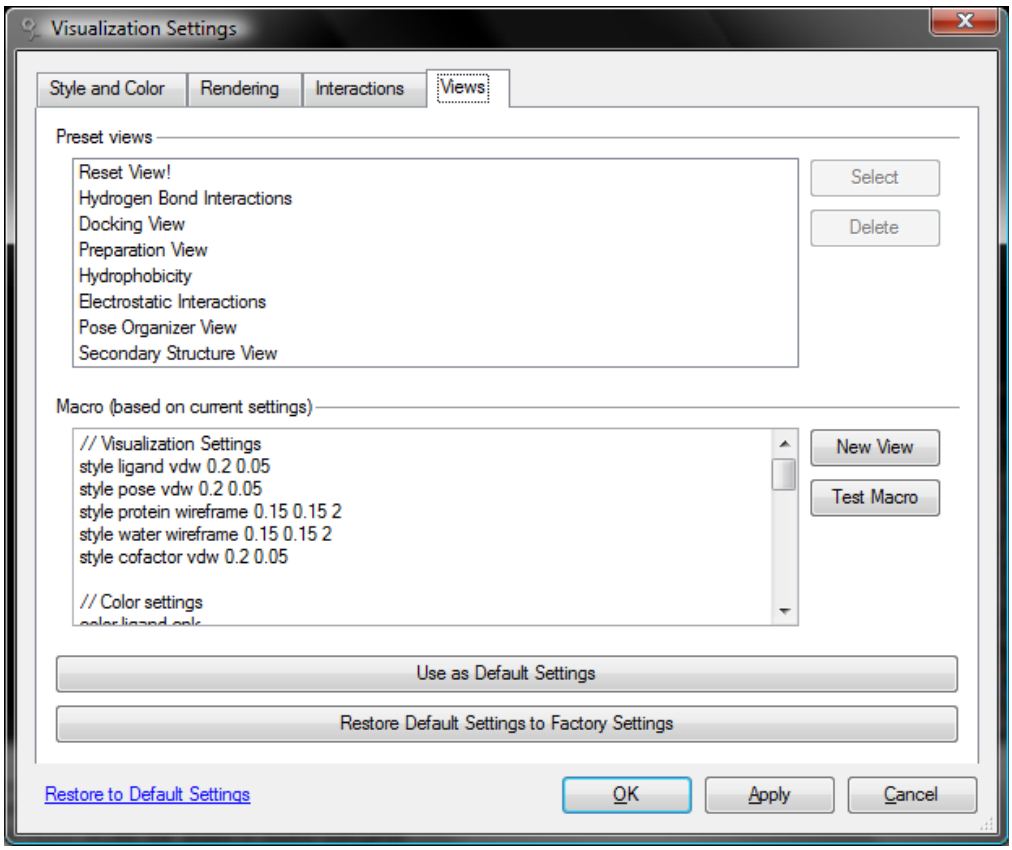


Figure 44:The Visualization Settings Views tab.

The lower panel allows you to create new views based on the current visualization settings. By pressing **New View** a dialog allows you to specify the name for the new view, after which it is added to the list of views on the main window menu bar. Views are stored as parts of the macros.xml file and appear under the **View** menu item.

It is also possible to modify the macro in the text-area before committing it as a macro. Modified macros can be tested by pressing **Test Macro** before they are stored permanently. It is possible to edit existing views in the **Macro and**

Menu Editor...

The default visualization settings used by MVD can be changed by pressing the **Use as Default Settings** button.

If needed, the default visualization settings can also be restored to the factory settings by pressing the **Restore Default Settings to Factory Settings** button. The factory settings are the initial settings used by MVD when started for the first time. At that point the factory settings are also used as the default visualization settings.

The current visualization settings shown in the Visualization Settings dialog will be stored in the MVDML workspace file when saving the workspace. When importing workspaces containing visualization settings, these stored settings will be used instead of the default settings.

Notice: When making a new workspace or clearing the current workspace, the default visualization settings will be used.

3.23 High-Quality Rendering

It is possible to create high-quality screenshots by selecting **Rendering** | **High-Quality Render (Raytrace)**

The High-Quality Render (Raytrace) dialog makes is possible to create images in arbitrary size and higher quality than when saving screenshots from the OpenGL view. The High-quality render uses a raytrace engine to create the output image. This has some graphical advantages as compared to the default OpenGL rendering: for instance spheres are not converted into triangle meshes before being drawn, and it possible to create shadow effects. Since another rendering technique is used, the output may deviate from the OpenGL view. The High-Quality Render also makes it possible to create high resolution images suitable for publications.

Notice, that a few graphical objects are not supported by the raytracer: dot surfaces, protonation guides, and energy grids. The raytracer also ignores clipping planes, and the light source settings in the Visualization Settings Dialog.

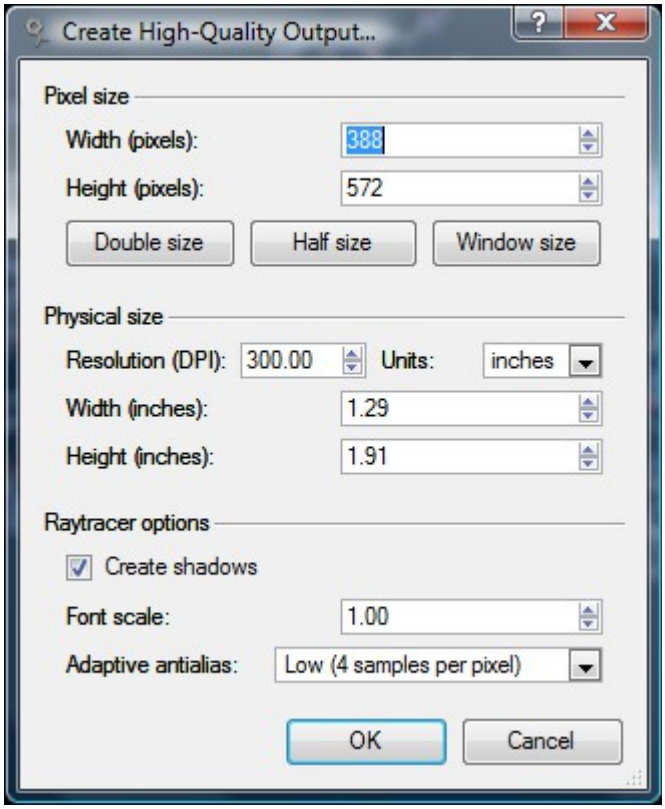


Figure 45: The High-Quality Output dialog.

The High-Quality Output dialog controls the size and rendering options. It is possible to specify an image size in either pixels or physical units. In order to use physical units, it is necessary to specify the printing resolution of the physical media - the default resolution is 300 DPI (dots per inch). It is possible to choose between inches and cm as units (but the DPI is always specified in inches).

Shadows can be toggled on and off, and it is possible to specify a font scale: since text is drawn differently by the raytracing engine, text may appear either too large or too small. This can be adjusted using the font scale settings. Adaptive antialias is a technique for reducing jarred boundaries between objects. Higher settings produce higher quality, but takes longer time to render.

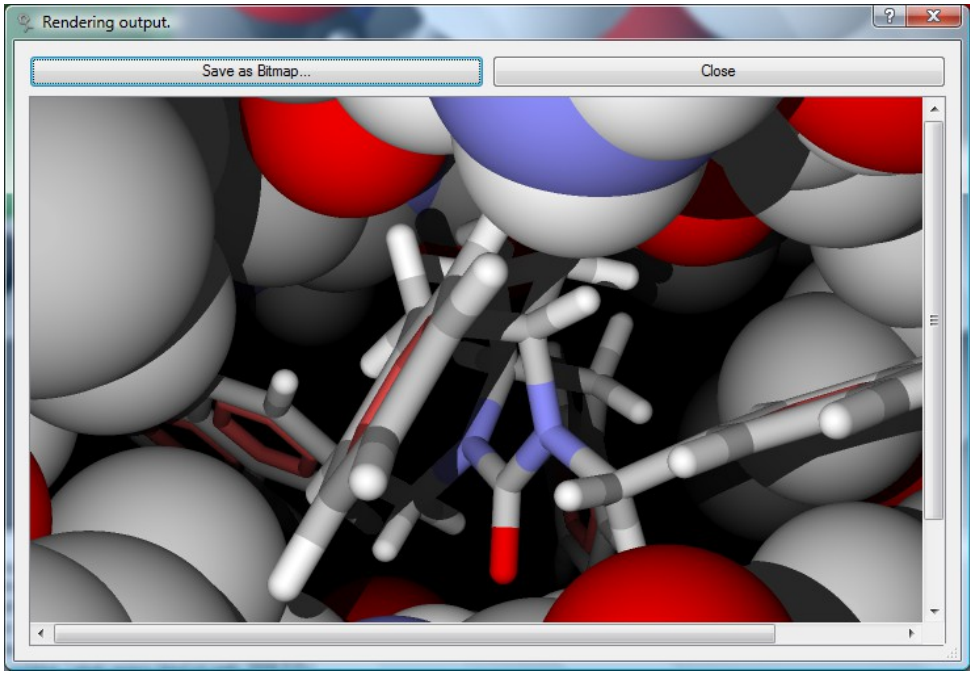
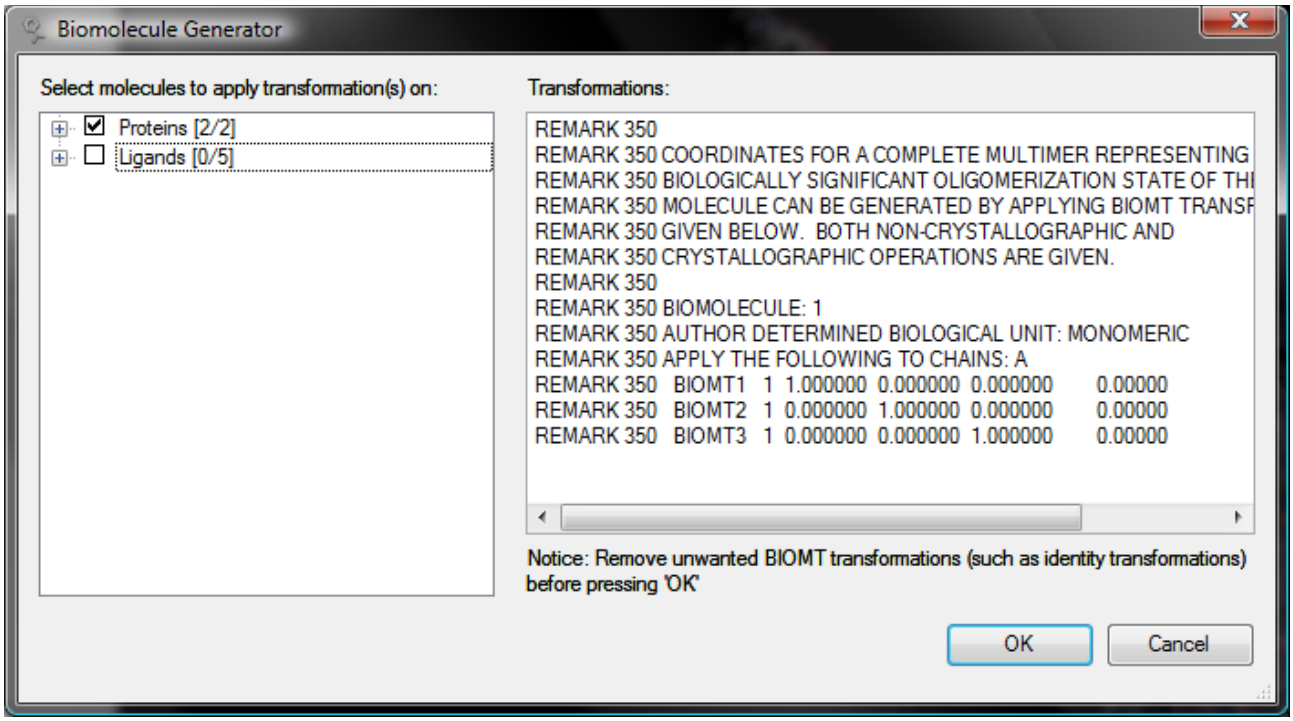


Figure 46: The output preview window.

After the output has been rendered, a preview window appears with the result, and the output can be saved as a bitmap. The PNG format produces the highest-quality images, since it uses loss-less compression, while the JPG format produces the smallest file sizes.

3.24 Biomolecule Generator

Some PDB files contain transformation information for generating biomolecules. To apply these transformations, invoke the **Biomolecule Generator** by choosing **Tools** | **Biomolecule Generator**.


Biomolecule Generator
Select molecules to apply transformation(s) on:
Proteins [2/2]
Ligands [0/5]
Transformations:
REMARK 350
REMARK 350 COORDINATES FOR A COMPLETE MULTIMER REPRESENTING
REMARK 350 BIOLOGICALLY SIGNIFICANT OLIGOMERIZATION STATE OF THI
REMARK 350 MOLECULE CAN BE GENERATED BY APPLYING BIOMT TRANSF
REMARK 350 GIVEN BELOW. BOTH NON-CRYSTALLOGRAPHIC AND
REMARK 350 CRYSTALLOGRAPHIC OPERATIONS ARE GIVEN.
REMARK 350
REMARK 350 BIOMOLECULE: 1
REMARK 350 AUTHOR DETERMINED BIOLOGICAL UNIT: MONOMERIC
REMARK 350 APPLY THE FOLLOWING TO CHAINS: A
REMARK 350 BIOMT1 1 1.000000 0.000000 0.000000 0.00000
REMARK 350 BIOMT2 1 0.000000 1.000000 0.000000 0.00000
REMARK 350 BIOMT3 1 0.000000 0.000000 1.000000 0.00000
Notice: Remove unwanted BIOMT transformations (such as identity transformations) before pressing 'OK'
OK
Cancel
Figure 47: The Biomolecule Generator.

The left panel on the dialog controls which molecules the transformation should be applied to. This is normally the proteins (or protein chains), but ligands, water and cofactors can also be transformed.

The right panel contains a text box where a transformation description can be pasted. Notice that if a transformation remark was present in the last loaded PDB file it will automatically appear here.

It can be necessary to manually edit the transformation remarks. For instance the remarks may contain redundant identity transformations which should be removed:

```
// Example of identity transformation.

REMARK 350   BIOMT1   1  1.000000  0.000000  0.000000        0.00000
REMARK 350   BIOMT2   1  0.000000  1.000000  0.000000        0.00000
REMARK 350   BIOMT3   1  0.000000  0.000000  1.000000        0.00000
```

PDB transformation remarks are triplets of remark lines, named BIOMT1-3. The first three columns constitute a rotation matrix, and the last column is a translation vector.

For some complex structures the transformation description may contain several steps where different transformations are applied to different subsets of the molecules. In this case it is necessary to run the **Biomolecule Generator** multiple times.

Also notice that biomolecules can be very large. Always render the protein in wireframe before attempting to generate large biomolecules.

3.25 Structural Alignment of Proteins

It is possible to structurally align proteins in Molegro Virtual Docker.

A structural alignment is done by matching a number of residues in two proteins and calculating the translation and rotation that minimizes the RMSD between the alpha-carbons in the matched residues.

The **Structural Protein Alignment** dialog can be invoked by selecting **Tools** | **Structural Protein Alignment** from the main menu.

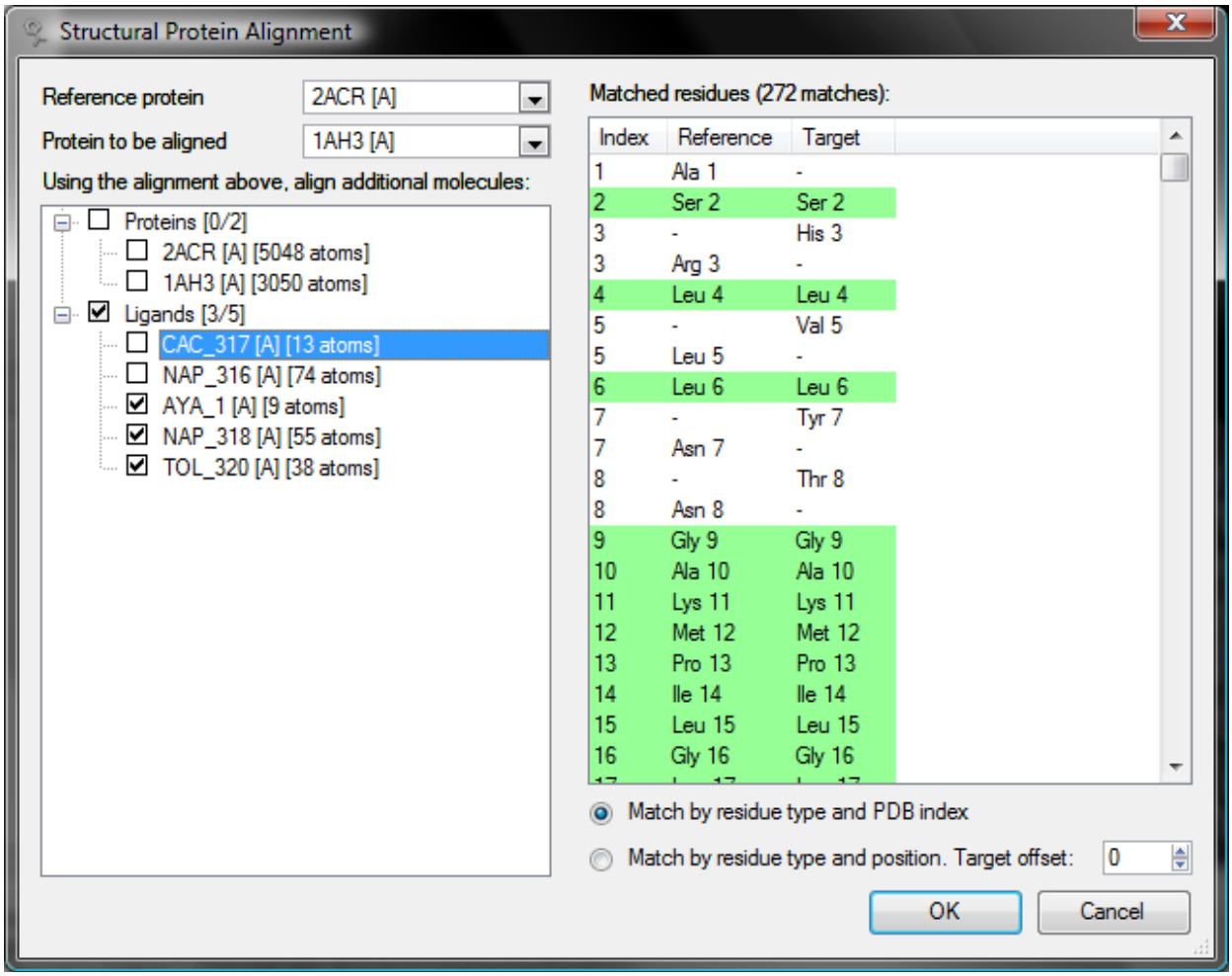

Structural Protein Alignment
Reference protein
2ACR [A]
Protein to be aligned
1AH3 [A]
Using the alignment above, align additional molecules:
Proteins [0/2]
2ACR [A] [5048 atoms]
1AH3 [A] [3050 atoms]
Ligands [3/5]
CAC_317 [A] [13 atoms]
NAP_316 [A] [74 atoms]
AYA_1 [A] [9 atoms]
NAP_318 [A] [55 atoms]
TOL_320 [A] [38 atoms]
Matched residues (272 matches):
Index Reference Target
1 Ala 1 -
2 Ser 2 Ser 2
3 - His 3
3 Arg 3 -
4 Leu 4 Leu 4
5 - Val 5
5 Leu 5 -
6 Leu 6 Leu 6
7 - Tyr 7
7 Asn 7 -
8 - Thr 8
8 Asn 8 -
9 Gly 9 Gly 9
10 Ala 10 Ala 10
11 Lys 11 Lys 11
12 Met 12 Met 12
13 Pro 13 Pro 13
14 Ile 14 Ile 14
15 Leu 15 Leu 15
16 Gly 16 Gly 16
Match by residue type and PDB index
Match by residue type and position. Target offset:
0
OK
Cancel

Figure 48: The Structural Protein Alignment dialog box.

The first step is to choose a reference protein and a protein to be aligned (the target protein). The target protein is the protein which will be translated and re-oriented.

When two proteins have been chosen, the list on the right side of the dialog will suggest a matching between residues in the proteins. Green entries indicate which residues that will be aligned. By default the matching will be done using **Match by residue type and PDB index** – where two residues will

be matched if they are of the same kind and have identical PDB residue identifiers.

Two PDB crystal-structures may have similar sequences, but different PDB residue identifiers. In this case it is possible to **Match by residue type and position.** This will match two residues if their positions in the sequences are identical. It is also possible to add a index offset to the target protein index.

Sometimes a number of other molecules are associated with a protein (a bound ligand or cofactor, or another protein chain). It is possible to select a number of additional molecules and apply the same transformation that aligns the target protein to the reference protein to the additional molecules. This is done by checking the desired molecules in the workspace view on the left side of the dialog. Notice that if the reference or target protein is selected as part of an additional alignment they will be ignored (since they are already considered).

3.26 Structural Alignment of Small Molecules

Simple alignment of small molecules is also possible. By selecting three atoms in one ligand, and selecting three atoms in another ligand, a new context menu appears when clicking on an atom in one of the molecules - **Align...**. This will align the molecules. The atoms are aligned in the same order as they are selected, that is, the first selected atom in ligand 1 is aligned to the first selected atom in ligand 2 etc. Therefore, it is important to ensure that the selection order is correct and that no other atoms are selected.

Notice: Only alignments with three selected atoms in each molecule are possible.

3.27 Macro and Menu Editor

The **Macro and Menu Editor** allows the user to modify existing menu entries or to extend the functionality by adding new menu entries. It can be invoked by choosing **Edit** | **Macro and Menu Editor**.

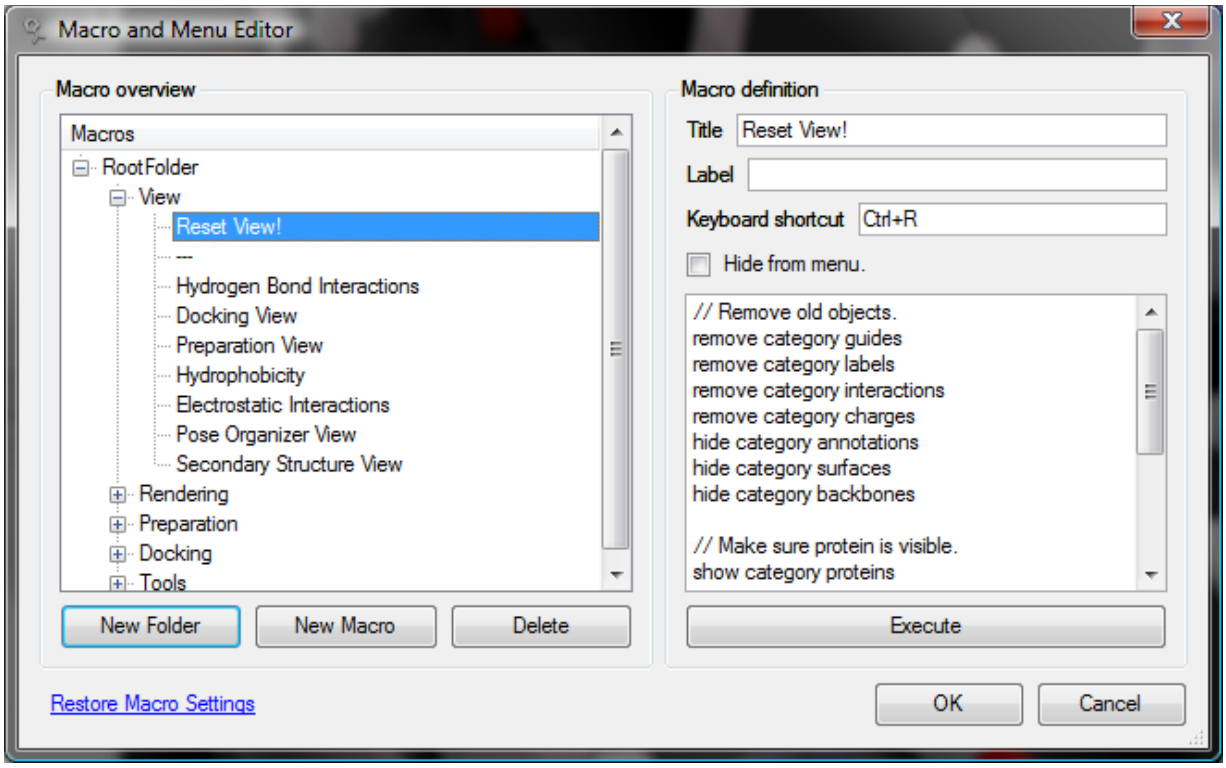


Figure 49: The Macro and Menu Editor.

The left pane (Macro overview) displays a hierarchical view of all macros. The top level folders are mapped directly to corresponding menus in MVD. That is, View, Rendering, Preparation and Docking will appear as menus in the GUI. It is possible to add new top-level folder, by selecting the root node (RootFolder) and pressing the New Folder button.

When a folder is highlighted in the **Macro overview**, new macros can be added to it, by pressing the **New Macro** button.

New or existing macros can be modified in the right pane (**Macro definition**).

A macro consists of a **Title**, which is the name that is shown in the corresponding menu, an optional **Label** which can be used to assign an unique name to the macro, so that it can be called from other macros (this is done by using the macro invoke command, i.e. '!macroname'), an optional **Keyboard shortcut** (which is specified as text i.e. 'Alt+F1' or 'Ctrl+Shift+1,Shift+A' where the last shortcut simultaneously maps two alternative keyboard shortcuts) and the actual **Macro definition**.

If macros or folders appear in red in the **Macro overview**, it is because **Hide from menu** is enabled for them. These items won't show up in the menus. This can be useful for defining macros which will not show up in the GUI, but still can be called from the **Console Window**.

It is also possible to add separators between the macros which will appear as menu separators in the GUI. To add a separator between macros just use ---

(3 strokes) as the **Title** of the macro. Similarly, separators can be created between macro folders. Again just use --- as the **Title** of the macro folder.

Macros can also be rearranged (e.g. changing the order of occurrence within a macro folder or moving macros between folders) by dragging and dropping a macro or a macro folder in the **Macro overview** listview.

If some macros are deleted or modified by mistake, the default macro settings can be restored by pressing the **Restore Macro Settings** link (located in the lower left corner of the dialog). Alternatively, the `macros.xml` file can be replaced by a backed-up version containing the default settings (`macros.backup`). Both files are located in the `Data` directory.

The actual commands that can be used to define the macros are described in Appendix X: Console and Macro Commands.

3.28 PDB and SDF Import Notes

When importing molecules from PDB or SDF files header and annotation information is stored as part of the current workspace. For PDB files the header is stored. For SDF files the first 4 lines and any annotations are stored.

Imported notes can be shown using the context menu on any molecule in the Workspace Explorer or by selecting a molecule in the Workspace Explorer and pressing the **Show PDB Header** or the **Show SDF Header** button for PDB and SDF files respectively.

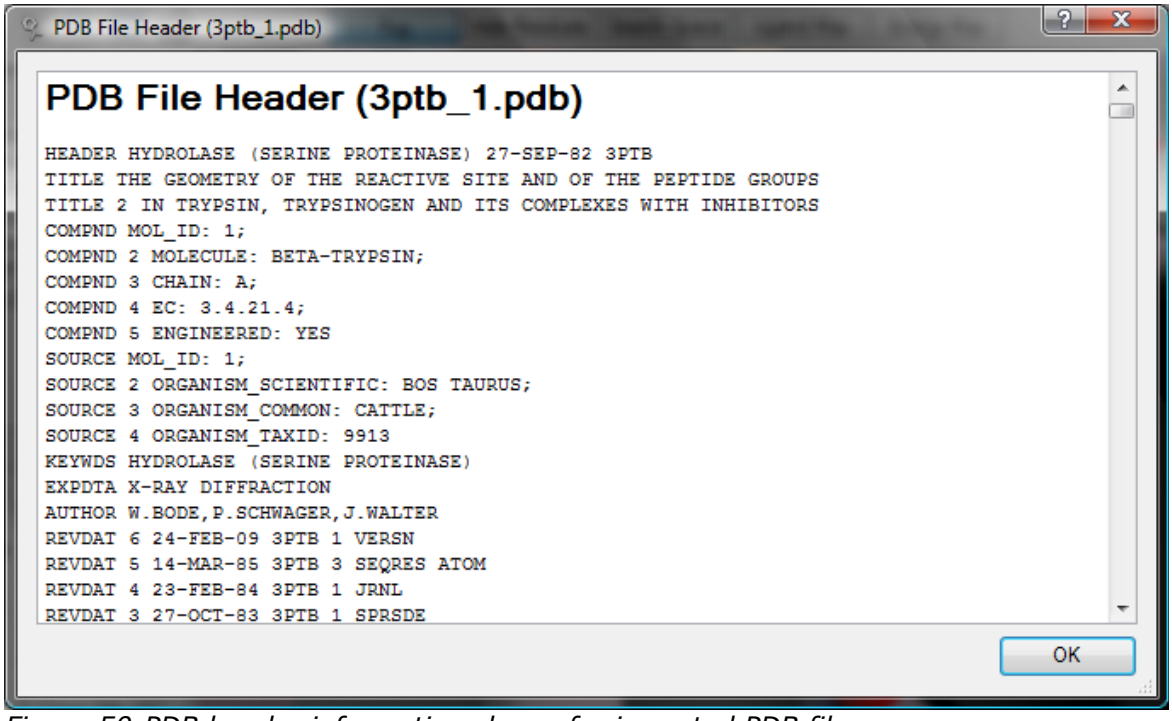

PDB File Header (3ptb_1.pdb)
PDB File Header (3ptb_1.pdb)
HEADER HYDROLASE (SERINE PROTEINASE) 27-SEP-82 3PTB
TITLE THE GEOMETRY OF THE REACTIVE SITE AND OF THE PEPTIDE GROUPS
TITLE 2 IN TRYPSIN, TRYPSINOGEN AND ITS COMPLEXES WITH INHIBITORS
COMPND MOL_ID: 1;
COMPND 2 MOLECULE: BETA-TRYPSIN;
COMPND 3 CHAIN: A;
COMPND 4 EC: 3.4.21.4;
COMPND 5 ENGINEERED: YES
SOURCE MOL_ID: 1;
SOURCE 2 ORGANISM_SCIENTIFIC: BOS TAURUS;
SOURCE 3 ORGANISM_COMMON: CATTLE;
SOURCE 4 ORGANISM_TAXID: 9913
KEYWDS HYDROLASE (SERINE PROTEINASE)
EXPDTA X-RAY DIFFRACTION
AUTHOR W.BODE,P.SCHWAGER,J.WALTER
REVDAT 6 24-FEB-09 3PTB 1 VERSN
REVDAT 5 14-MAR-85 3PTB 3 SEQRES ATOM
REVDAT 4 23-FEB-84 3PTB 1 JRNL
REVDAT 3 27-OCT-83 3PTB 1 SPRSDE
OK

Figure 50:PDB header information shown for imported PDB file.

A workspace may contain an arbitrary number of import notes, and each molecule may have a reference to one of these notes.

Imported notes are stored in the MVDML workspace file and they can be viewed and deleted using the Workspace Properties dialog.

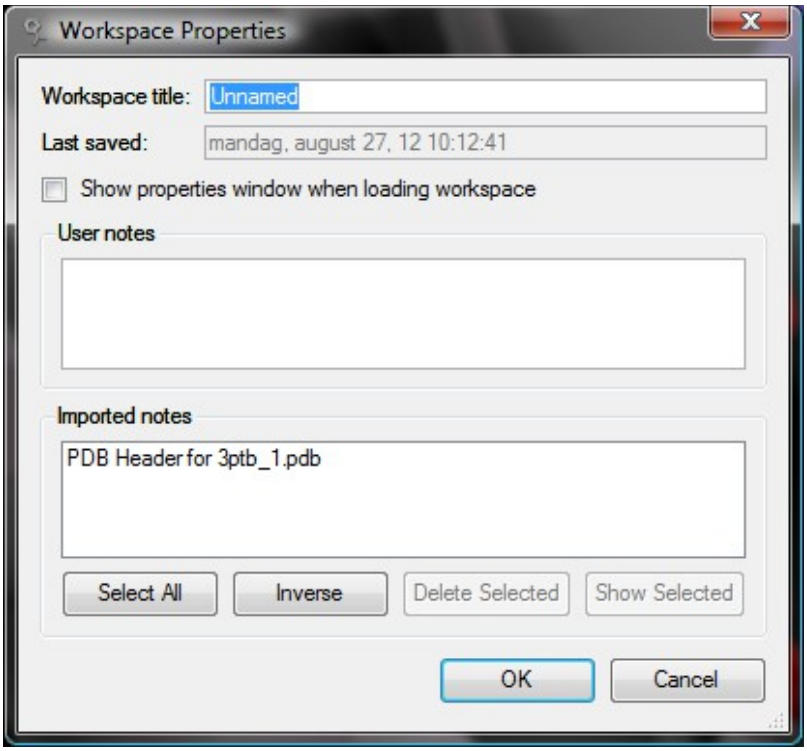


Figure 51:Imported PDB and SDF notes can be shown and deleted using the Workspace Properties dialog.

Notes that are no longer referenced by a molecule are automatically removed.

3.29 Energy Maps

It is possible to visualize the force fields in Molegro Virtual Docker using the **Energy Map Visualization** dialog. The **Energy Map Visualization** dialog can be invoked by pressing the **Energy Map** button on the main GUI toolbar.

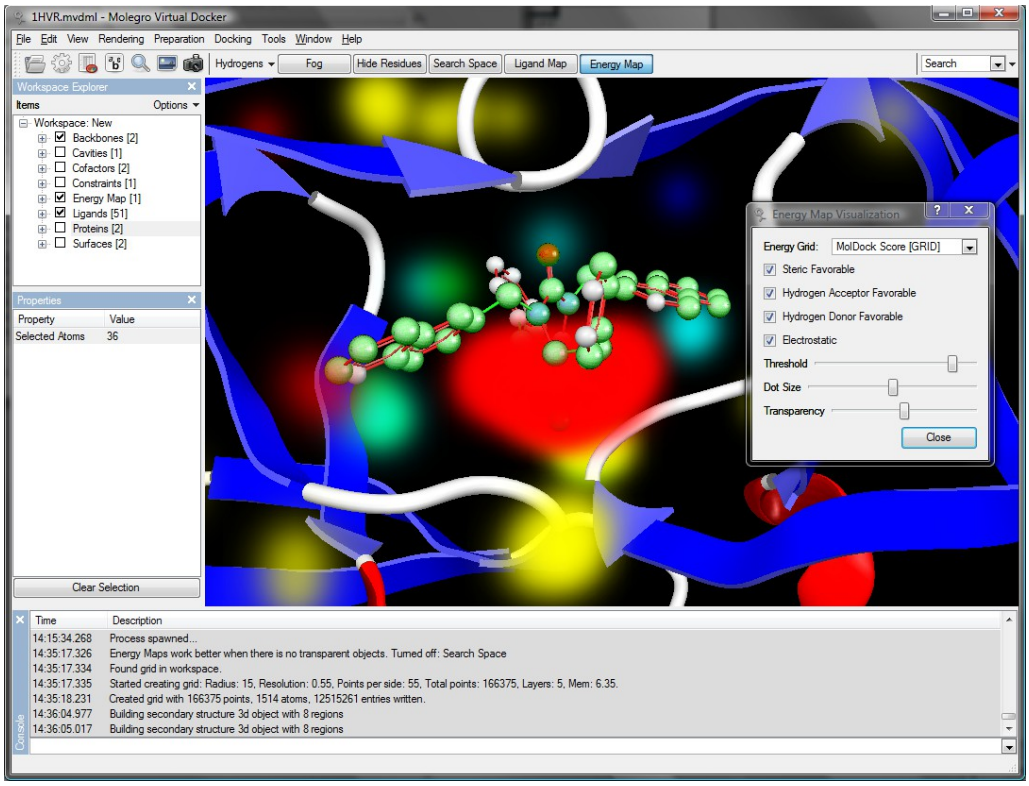


Figure 52: Energy Grid Visualization

The "MolDock [GRID]" and "PLANTS Score [GRID]" scoring functions use a precalculated energy grid during the docking simulations to accelerate the protein-ligand interaction calculations.

Visualising these potential fields makes it possible to gain an understanding of which regions are attractive to the atoms in a ligand. Four different types of energy potentials can be shown:

- **Steric Favorable**. These are the regions where it is favorable to place non-polar atoms. These volumes are visualized in green. This field will be strongest near the surfaces and in cavities.
- **Hydrogen Acceptor Favorable**. These are the spots where it is favorable to place a ligand atom capable of accepting a hydrogen bond, i.e. spots near a hydrogen donor in the protein. They appear as blue regions. Notice, that the fields do not take direction into account (for instance, the position of the hydrogen in a hydroxyl group is not taken into account, when calculating the field - it is assumed to be able to point in any direction).
- **Hydrogen Donor Favorable.** The regions show favorable spots for heavy atoms in the ligand that are able to donate a hydrogen to a hydrogen bond. They appear in yellow.
- **Electrostatic**. Show the electrostatic potential of the protein. Red

regions correspond to a nearby negative electro-static charge in the protein. Blue regions correspond to a nearby positive charge in the protein. The electrostatic potential is the sum of the Coulomb potentials for each atom in the protein, with a distance-dependent dielectric constant (see Chapter 18). The electrostatic field is not used with the PLANTS Scoring function.

- All the fields take the steric interaction into account, so that only grid positions where it is possible to place an atom without steric clashes with the protein are shown. For instance, there may be charged regions inside the protein which are not shown, because it is not possible to place a ligand atom there.

It is possible to adjust the appearance of the fields:

- The **Threshold** slider determines at which point an interaction is strong enough to be included in the visualization. A higher threshold value results in less points being shown.
- The **Dot Size** determines how large the transparent blobs that make up the volumetric fields are.
- The **Transparency** slider controls how opaque the fields appear. Higher values make the fields more opaque.

4 Preparation

4.1 Import of Molecules

Molecules can be imported into MVD using the **Import Molecule...** menu option located in the **File** menu. A shortcut is provided from the tool bar by clicking on the **File** folder icon or using the **Ctrl-O** keyboard shortcut. Molecules can also be imported by dragging-and-dropping the molecular file into the main application window.

Currently, MVD supports the following file formats:

- Protein Data Bank (pdb/ent)
- Sybyl Mol2 (mol2)
- MDL (sdf/sd/mol/mdl)

Notice that only PDB and Mol2 files can contain proteins, and water molecules. In general, it is recommended to use Mol2 or SDF files for ligands since they can contain bonding information.

From the **Import Molecules** dialog shown in Figure 53, it is possible to select which molecules to import, prepare molecules, and inspect warnings found during parsing of the imported file.

Notice: If more than 10 ligands are present in the file (typically SDF or Mol2 files), a subset of the ligands can be selected for import using the **Specify ligand range** option (see Figure 53). Since it is computationally slow to display a large number of molecules (e.g. thousands of compounds), ligands and poses are not automatically shown in the **Visualization Window** if the number of molecules imported exceeds 50 (for each category).

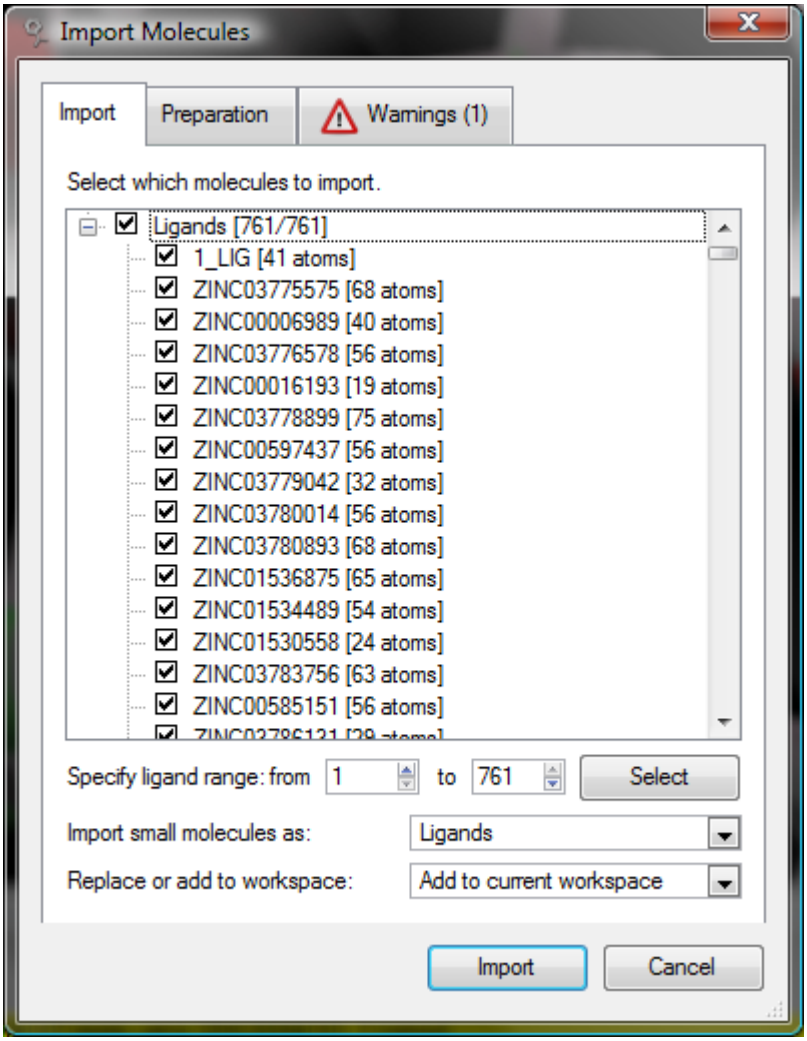


Figure 53: Import Molecules dialog.

When all relevant molecules have been imported, the molecules can be automatically prepared (see next section).

MVD automatically tries to identify cofactors: a molecule is considered a cofactor if it has less than 5 heavy atoms or its name is included in a list of common cofactor names (like 'HEM', 'SO4', 'PO4', ...). If this is not desired, it is possible to override cofactor recognition by checking the **Import cofactors as ligands** option.

4.2 Automatic Preparation

Some molecular file formats support information about bond type and charge (e.g. Mol2) while others do not (e.g. PDB). In order to maker proper predictions, it is important that the structures have been properly prepared. That is, that the atom connectivity is known and that the correct bond order and charges have been assigned.

The **Prepare Molecules** dialog allows the user to perform the necessary

preparation. It is invoked automatically when importing Mol2, SDF, or PDB files, and can be invoked manually by selecting **Preparation** | **Prepare Molecules** or by using the context menu (e.g. **Prepare Ligand...**) on molecules in the **Workspace Explorer**.

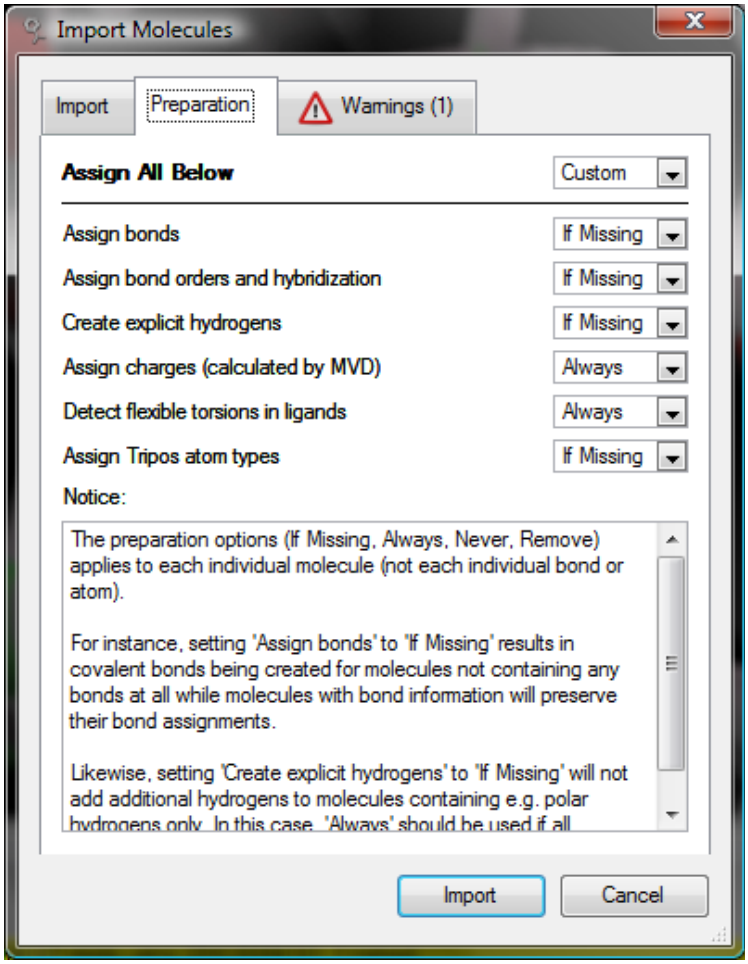

Import Molecules
Import
Preparation
Warnings (1)
Assign All Below
Custom
Assign bonds
If Missing
Assign bond orders and hybridization
If Missing
Create explicit hydrogens
If Missing
Assign charges (calculated by MVD)
Always
Detect flexible torsions in ligands
Always
Assign Tripos atom types
If Missing
Notice:
The preparation options (If Missing, Always, Never, Remove) applies to each individual molecule (not each individual bond or atom).
For instance, setting 'Assign bonds' to 'If Missing' results in covalent bonds being created for molecules not containing any bonds at all while molecules with bond information will preserve their bond assignments.
Likewise, setting 'Create explicit hydrogens' to 'If Missing' will not add additional hydrogens to molecules containing e.g. polar
Import
Cancel

Figure 54: Preparing molecules.

Within all preparation types the following four different possibilities are available (see Figure 54):

- **Always**. Unconditionally performs the preparation by MVD.
- **Never**. Skips the preparation.
- **If Missing**. The preparation will only be performed if no knowledge is already present (e.g. if bond orders exist in the Mol2 file, bond orders are not assigned by MVD. However, if bond order information is not included, MVD will assign it).
- **Remove**. Tries to remove preparation (e.g. if 'Assign bond orders...' is set to 'remove', all bond orders will be set to single bonds. If 'Create explicit hydrogens' is set to 'remove' all hydrogen atom are removed).

Notice: The preparation options (Always, Never, If Missing, Remove) applies to each individual molecule (not each individual bond or atom). For instance, setting 'Assign bonds' to 'If Missing' results in covalent bonds being created for molecules not containing any bonds at all while molecules with bond information will preserve their bond assignments. Likewise, setting 'Create explicit hydrogens' to 'If Missing' will not add additional hydrogens to molecules containing e.g. polar hydrogens only. In this case, 'Always' should be used if all hydrogens should be created.

Assign Bonds

This option allows to determine which atoms are connected (covalently bound). Two atoms are connected if their distance is more than 0.4Å and less than the sum of their covalent radii plus a threshold of 0.45Å (the threshold is set to 0.4865Å if one of the atoms is Phosphorus).

Assign Bond Order and Hybridization

This options allows recognition of bond orders (whether bonds are single, double or triple, ...), the number of hydrogens attached to the atoms, and their hybridization (SP, SP2, SP3). Also aromatic rings will be detected. It should be noted that this assignment is not always perfect - different protonation states can be difficult to assign properly. A detailed description can be found in Appendix VII: Automatic Preparation.
Notice: The algorithm only assigns the number of implicit hydrogens to each atom. No actual atoms will be added. The next option **Create explicit hydrogens** allows you to add explicit hydrogens based on the implicit ones.

Create Explicit Hydrogens

Creates hydrogens matching the predicted number of hydrogens in the step above. The hydrogens are placed according to geometric criteria (i.e. SP3 hybridized atoms are kept at a 109 degrees geometry). The hydrogens are placed at standard distances according to the atom they are connected to. No energy minimization is performed.

Assign Charges

This option allows to assign partial charges to each atom based on the scheme described in Appendix I: MolDock Scoring Function.

Detect Flexible Torsions In Ligands

This option determines which bonds that should be considered flexible during docking. It is advisable always to set this option to either **If Missing** or

Always. If this option is set to **Remove**, the ligand will be considered rigid during docking.

Assign Tripos Atom Types

This option is used to assign Tripos atom types using a built-in heuristic. If the option is set to **Never**, atom types will be imported from the molecule file, instead of being assigned by MVD (only available for Mol2 structural files). The **Remove** option will set all atom types to 'Undefined'. **Always** will assign Tripos atom types to all atoms using built-in assignment rules, and **If Missing** (default) will assign atom types to 'Dummy', 'Undefined' and 'Other' typed atoms using built-in rules (all other atom types will be imported from the Mol2 file).

Hydrogen Bonding Type

Atom hydrogen bonding types (acceptor, donor, both or non-polar) are always set during preparation.

4.3 Manual Preparation

Molecules can be manually prepared using the context menus of highlighted atoms or bonds (see below).

Set Hybridization

Hybridization (SP, SP2, SP3) can be manually assigned to atoms by right-clicking on the atom in question and selecting the **Set Hybridization** menu option.

Set Hydrogen Bond Type

The hydrogen bond type used by MolDock scoring function (donor, acceptor, both, non-polar) can be manually assigned to atoms by right-clicking on the atom in question and selecting the **Set Hydrogen Bond Type** menu option.

Set Tripos Atom Type

Sometimes, the built-in assignment scheme fails in assigning correct Tripos atom types to specific atom. In such cases, it is possible to change the Tripos atom type for nitrogen, oxygen, carbon, and sulphur atoms by right-clicking on the atom in question and selecting the **Set Tripos Atom Type** menu option.

Set Plants Atom Type

By default, MVD automatically assigns Plants atom types (Donor, Acceptor,

Both, Nonpolar, Metal) before docking with PLANTS Score using the rules described in [KORB 2009]. However, it is also possible to manually assign the Plants atom type by right-clicking on the atom in question and selecting the **Set Plants Atom Type** menu option. Notice: Plants atom types are not defined for hydrogen atoms.

Set Hydrogen Count

The **Set Hydrogen Count** menu option can be used to set the number of explicit hydrogens attached to the highlighted atom.

Assign Charges

The MolDock scoring function uses partial charges assigned when running the **Preparation** dialog. However, the assignment of charges is based on standard templates and charge assignments can be missing in some cases. It is possible to manually assign partial charges to atoms by right-clicking on the atom in question and selecting the **Set Partial Charge** menu option.

Set Bond Order

Bond orders can be manually assigned by right-clicking on the bond in question and selecting the **Set Bond Order** menu option.

Notice that bonds are not visible in some visualization styles. The most suitable view is the ball-and-stick style, which can be set from the **Rendering** menu in the menu bar.

Set Ligand Flexibility

Flexible torsions in the ligand can manually be set rigid or flexible by right-clicking on a bond and selecting the **Set Flexibility** menu option.

Set Root Atom

When automatically detecting and assigning flexible torsion angles (using the automatic preparation procedure), a root atom is chosen. The root atom is used as root in the torsion tree, which is used to construct the ligand conformation during the docking process. Sometimes, the docking performance can be improved by choosing another atom to be the root atom. To manually set the root atom, right-click on an atom and select the **Set as Root Atom** menu option.

Notice that bonds are not visible in some visualization styles. The most suitable view is the ball-and-stick style, which can be set from the **Rendering** menu in the menu bar.

4.4 Protein Preparation

The **Protein Preparation** dialog allows you to inspect the proteins in the workspace for structural errors (such as missing atoms or erroneous bonds) and to inspect and change the protonation state for the residues. It is also possible to mutate residues (for instance replacing an asparagine residue with an aspartic acid residue) and subsequently energy minimize them.

The protein preparation dialog can be invoked by choosing **Preparation** | **Protein Preparation** from the main menu bar.

When the protein preparation dialog is invoked, a list of residues is shown. All residues with potential errors are initially highlighted on the list and emphasized in the 3D view with yellow or red spheres corresponding to the two different kinds of residue errors:

- Residues with structural errors. These kinds of residues do not match the atom and bond information in the *protonation templates* (explained below). They might have missing atoms or invalid bonds between the atoms. Notice that terminal residues does not always match the standard templates – they may contain additional atoms such as a terminal oxygens (OXT).
  It is not possible to change protonation for residues with structural errors – but they may be reconstructed by using the **Mutate and Optimize** tab to 'mutate' to a residue of the same type. These residues are shown with red spheres in the 3D view.
- Residues with a valid atomic structure, but with an invalid protonation. These kinds of errors occur if the residue does not match any of the defined protonation states for the given residue. These errors can be fixed by changing the protonation state into a valid state. These residues are shown with yellow spheres in the 3D view.

4.5 The Protonation Tab

On the protonation tab, a list view displays all residues for the proteins in the workspace. The green arrows jump to the next or previous erroneous residue (either improper structure or unknown protonation state).

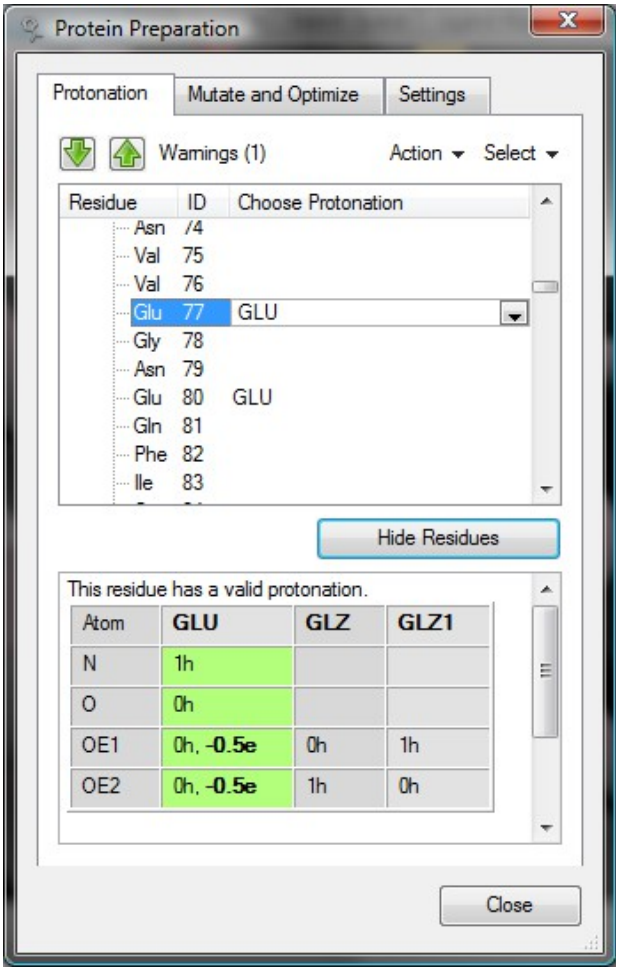


Figure 55: The Protein Preparation Dialog. The lower panel displays the current protonation state.

If a residue has alternate protonation states, they can be chosen in two different ways:

- By choosing a residue in the list view, and then changing the state from the drop-down menu in the third column ('Choose Protonation').
- By using the context menu in the 3D visualization view on a residue marked by a red or yellow sphere.

Because it can be difficult to get an overview of the individual residues, it is possible to invoke the **Hide Residues** dialog directly from the protonation preparation dialog. The **Hide Residues** button provides a shortcut for invoking this dialog - the functionality is the same as described in Section 3.9.

The text window at the bottom displays information about the currently selected residue. Here it is possible to see the different protonation states, and any errors may be inspected here.

Atom	GLU	GLZ	GLZ1
C	0h		
CA	1h		
CB	2h		
CD	0h		
CG	2h		
N	1h		
O	0h		
OE1	0h, **-0.5e**	0h	1h
OE2	0h, **-0.5e**	1h	0h

Figure 56: Example of protonation state. Some fields for the alternative protonations (GLZ and GLZ1) are blank. These blank fields must match the base protonation (GLU) in order for a residue to match an alternative protonation.

All protonation states consist of a *base* or *default protonation*, which describes the number of hydrogens for each atom, the bonding between atoms and their charge. The base protonation is listed as the first column in the table (in figure the base protonation is listed in the GLU column). Alternative protonations are modifications to this base scheme. In Figure 56 the GLZ and GLZ1 columns are modifications to the 'GLU' scheme – they provide only information for some of the atoms in the residue (in this case OE1 and OE2). For a residue to match an alternative protonation, the atoms must match the properties described by the alternative protonation, while any atom not described by the alternative protonation must match the base protonation.

Finally, the protonation tab also provides two drop-down menus. The first, **Action**, provides a single option: **Set All Unknown to Default Protonation**. Invoking this option sets all residues with an unknown protonation to their default protonation state. The second drop-down menu **Select** provides an easy way to select multiple residues. The following selections are possible: **Residues with Invalid Structure** selects all residues with structural errors (missing atoms or erroneous bonds). **Residues with Unknown Protonation** selects all residues with protonation schemes not matched by any of the residue templates. Finally the residues most likely to have a non-default protonation state (His, Glu, and Arg) can be selected using this drop down menu.

Notice that the protonation templates are user-customizable. See the last section 'Customizing the protonation templates' for more information.

4.6 The Mutate and Optimize Tab

In order to mutate/change the residue type, select a single residue from the list (it is not possible to mutate multiple residues at once). Whenever a new residue is chosen from the 'Mutate to:' drop-down list, the sidechain is replaced and the 3D view is updated to reflect the changes.

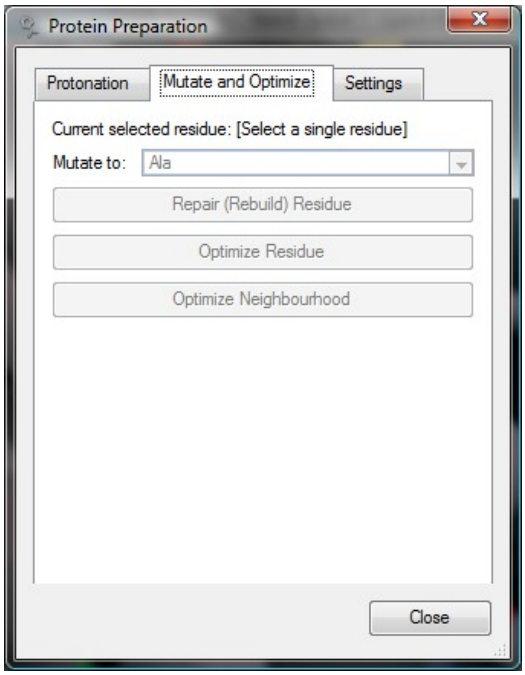


Figure 57: The residue mutation and optimization tab.

Residues are substituted by taking the corresponding template defined in the '`misc/data/residuetemplates.mvdml`' file, and aligning the N, C, and CA backbone atoms with the N, C, and CA backbone atoms for the chosen residue.

After having substituted the new residue it is recommended to optimize its position. A quick optimization can be performed by choosing **Optimize Residue**. This will perform a search for the best dihedral angles for the residue. It is also possible to optimize the positions of neighbouring residues as well. The optimization uses the same approach as the Sidechain Minimization dialog which is described in Section 3.20.

By choosing **Optimize Neighbours** both the chosen residue and the residues which are closest to it are selected. The selection is done as follows: each residue is assigned a bounding sphere (a sphere which is large enough to enclose the residue in all possible conformations). If the distance between two residues are less than the threshold distance specified in the settings tab (**Residue neighbour distance (Å)**), the residues are considered to be neighbours. By default the neighbour distance is 0 Å, meaning that the bounding spheres must overlap for residues to be considered neighbours. Increasing the distance results in a larger neighbourhood.

4.7 The Settings Tab

The following settings can be customized:

Check (and correct) charges

If this option is checked, all atomic partial charges are validated against the values defined in the protonation template file. The charges defined in this file uses a deliberately simple charge scheme where only a few atoms are assigned charge (see Appendix I: MolDock Scoring Function). If, however, the receptor has been prepared in another program with another charge assignment scheme (and saved in a format such as Mol2 which supports partial atomic charges), uncheck this option in order not to receive several of warnings about wrong atomic partial charge.

Notice that this setting affects both the validation and how protonation states are changed. When a protonation state is changed from the protein preparation dialog, the charges are only modified if this setting is checked.

Check (and correct) non-polar atoms (carbons)

Some PDB files contain only explicit hydrogen information for the polar atoms. Since the hydrogen information for non-polar atoms (the carbons) is not used by the MolDock score during the docking it is not necessary to have them explicitly attached. Therefore, by default only the hydrogen count for polar atoms is checked.

As above this setting affects both the validation and how protonation states are changed. When a protonation state is changed from the protein preparation dialog, hydrogens on non-polar atoms are only modified if this setting is checked.

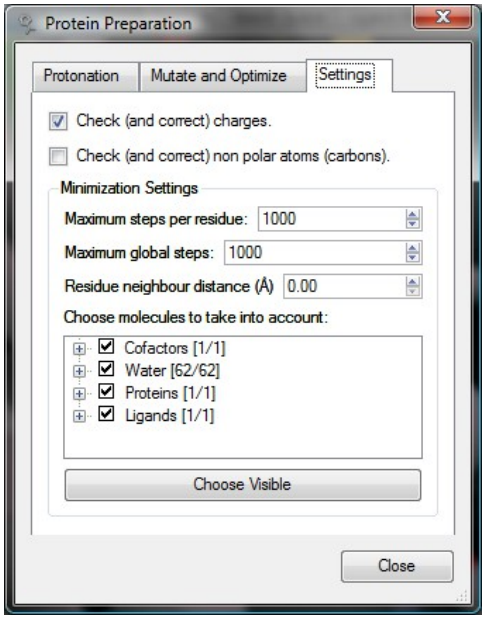


Figure 58: The Settings tab

Minimization Settings

The minimization options are the same as the ones described in the Sidechain Minimization section (see 3.20).

The only difference is the inclusion of the **Residue neighbour distance (Å)**, which determines how close residues must be in order to be considered neighbours (this criteria is described in the 'The Mutate and Optimize Tab' section).

4.8 Customizing the Protonation Templates.

The protonation templates are defined in the '`misc/data/residues.xml`' file.

It is possible to manually modify and extend this file with new protonation patterns, but we strongly advise that a backup copy of the original file is made before doing so. The protonation template file must be valid XML (Wikipedia offers an introduction to XML at `http://en.wikipedia.org/wiki/Xml`).

The overall structure of the XML file is illustrated with the following fragment from the protonation template file:

```
<ResidueDefinitions>
  ...
  ...
  <Residue name="ASP" letter="D" longName="Aspartate" pdbAlias="ASP">
    <Atom pdbName="C" hyb="2" charge="0" hydrogens="0" element="C" />
    <Atom pdbName="CA" hyb="3" charge="0" hydrogens="1" element="C" />
    <Atom pdbName="CB" hyb="3" charge="0" hydrogens="2" element="C" />
    <Atom pdbName="CG" hyb="2" charge="0" hydrogens="0" element="C" />
    <Atom pdbName="N" hyb="2" charge="0" hydrogens="1" element="N" />
    <Atom pdbName="O" hyb="2" charge="0" hydrogens="0" element="O" />
    <Atom pdbName="OD1" hyb="3" charge="-0.5" hydrogens="0" element="O" />
    <Atom pdbName="OD2" hyb="2" charge="-0.5" hydrogens="0" element="O" />
    <Bond from="CA" to="N" order="1"  />
    <Bond from="CA" to="CB" order="1"  />
    <Bond from="CA" to="C" order="1"  />
    <Bond from="C" to="O" order="2"  />
    <Bond from="CB" to="CG" order="1"  />
    <Bond from="CG" to="OD2" order="2"  />
    <Bond from="CG" to="OD1" order="1"  />
      <Protonation name="ASZ" pdbAlias="ASZ1" description="OD2 protonated (Neutral)">
        <Atom pdbName="OD1" charge="0" hydrogens="0" />
        <Atom pdbName="OD2" charge="0" hydrogens="1" />
    </Protonation>
```

```
      <Protonation name="ASZ1" pdbAlias="ASZ1" description="OD1 protonated (Neutral)">
        <Atom pdbName="OD1" charge="0" hydrogens="1" />
        <Atom pdbName="OD2" charge="0" hydrogens="0" />
   </Protonation>
  </Residue>
    <Bond from="CA" to="N" order="1"  />
    <Bond from="CA" to="C" order="1"  />
    <Bond from="C" to="O" order="2"  />
  </Residue>
  ...
  ...
<ResidueDefinitions>
```

The residue template always consists of a *base* or default protonation, which describes the atoms in the residue, their hybridization, element type, partial charge and their number of hydrogens. The base protonation also describes the bonds in the residue and the order of these bonds. The base protonation is described by the <Atom> and <Bond> elements that are immediate children of the <Residue> element.

A residue may also contain a number of *alternative protonations*. These are described by the <Protonation> elements. The alternate protonation are considered modifications to the base protonation, so they describe only the differences to the base protonation. Any <Atom> or <Bond> tag in an alternate protonation description will replace the settings inherited from the base protonation.

A short description of the various attributes is shown in the table below:

Element	Attributes
<Residue>	The 'name' attribute is used to identify the residues in a PDB file. The other attributes ('letter', 'longName' and 'pdbAlias') are purely informational and not used during parsing).
<Atom>	'pdbName' is used to identify the atom in the PDB file. 'hyb' describes the hybridization of the atom (2 = SP2 and 3 = SP3). 'Charge' is the atomic partial charge. 'Hydrogen' is the number of hydrogens attached to this atom. 'Element' is the element type.
<Bond>	'from' and 'to' must be 'pdbNames' of the atoms this bond connects. The 'order' attribute describes the bond order (1 = single bond, 2 = double bond, and 1.5 = delocalized bond).
<Protonation>	'name' refers to the name that will be used as display name and identifier in the GUI. 'pdbAlias' and 'description' are purely informational.

5 Data Sources

There are several ways to import ligands and prepare them for docking in Molegro Virtual Docker.

- Ligands can be imported in the GUI (using **Import Molecules...** from the **File** menu) and included in the workspace before docking. This is the easiest way to import data, but it can be slow if working with thousands of ligands.
- Ligands can be imported using the 'IMPORT' script commands. This has the disadvantage that all of the input file is parsed (e.g. a SDF-file containing 2000 entries will have to be completely loaded and prepared in memory, even if only a subset of it is needed). It is also necessary to modify the MVD-scripts manually.
- Ligands can be read from a *Data Source*. Ligands are 'streamed' from a source (such as a large file) and only one molecule is loaded into memory at a time.

Currently two types of data sources are available in Molegro Virtual Docker:

- *File data sources.* These are single files containing multiple structures (such as SDF, multi-molecule Mol2, or MVDML). It is possible to read a subset of the molecules contained in the file.
- *Multifile data sources*. These can be used when the input structures are split over several different files. A multifile data source may contain files with a mixture of different data formats.

5.1 Data Sources Syntax

File Data Sources

File data sources are identified by a 'File=' identifier. Examples:

```
File=\\fileserver\molecules\mol23.mol2
File="C:/Test Molecules/steroids.sdf";Index=2,4-8,12,34-
```

It is possible to import a subset of the structures in a file using the 'Index' specifier.

Molecules must be separated either by '$$$$' for SDF files or '@<TRIPOS>MOLECULE' for multi-molecule Mol2 files. Only one molecule will be extracted from each section separated by these separators. For PDB files only the first HETATM molecule will be imported.

Notices that all input structures are expected to be ligands. Molecules recognized as proteins or water molecules will be ignored.

The optional 'Index' specifier must be a comma-separated list of either single values or intervals. Notice that open intervals are allowed (e.g. '5-' or '-19'). Indices should be ordered strictly increasing. Invalid or non-existent indices will be ignored. The 'Index' specifier is 1-based (the number of the first molecule is 1 and not 0).

Filenames containing spaces must be enclosed in quotation marks. It is possible to specify files on shared network drives and folders.

Multifile Data Sources

Multifile data sources are identified by a 'Dir=' identifier. Examples:

```
Dir="C:/Test Molecules";Pattern="*.sdf;*.mol2";Index=10-100
Dir=C:/Test;Pattern=Stereo*.sdf;Index=10-100
```

The Multifile data source takes a directory and scans it for the given pattern. Patterns are specified using '*' as a wildcard. Notice that on Linux and Mac operating systems, file patterns are case sensitive.

It is possible to specify more than one pattern by separating sub-patterns with semi-colons. Patterns with semi-colons must be surrounded by quotes.

As with file data sources it is possible to specify a subset using the molecule index specifier ('Index'). Notice, that the 'Index' specifier refers to the molecule index – not the file index.

5.2 Using Data Sources

Data sources can be constructed and used in the following ways:

Specifying a Data Source in the Docking Wizard

The first page in the Docking Wizard (Choose Which Ligands To Dock) allows you to choose to dock from a data source.

Notice that it is not possible to specify an RMSD reference ligand when docking with data sources (since reference ligands must have compatible atoms and this cannot be checked for data sources).

The docking wizard creates a script where the DOCK command contains a reference to the specified data source (see 'using data sources from a script').

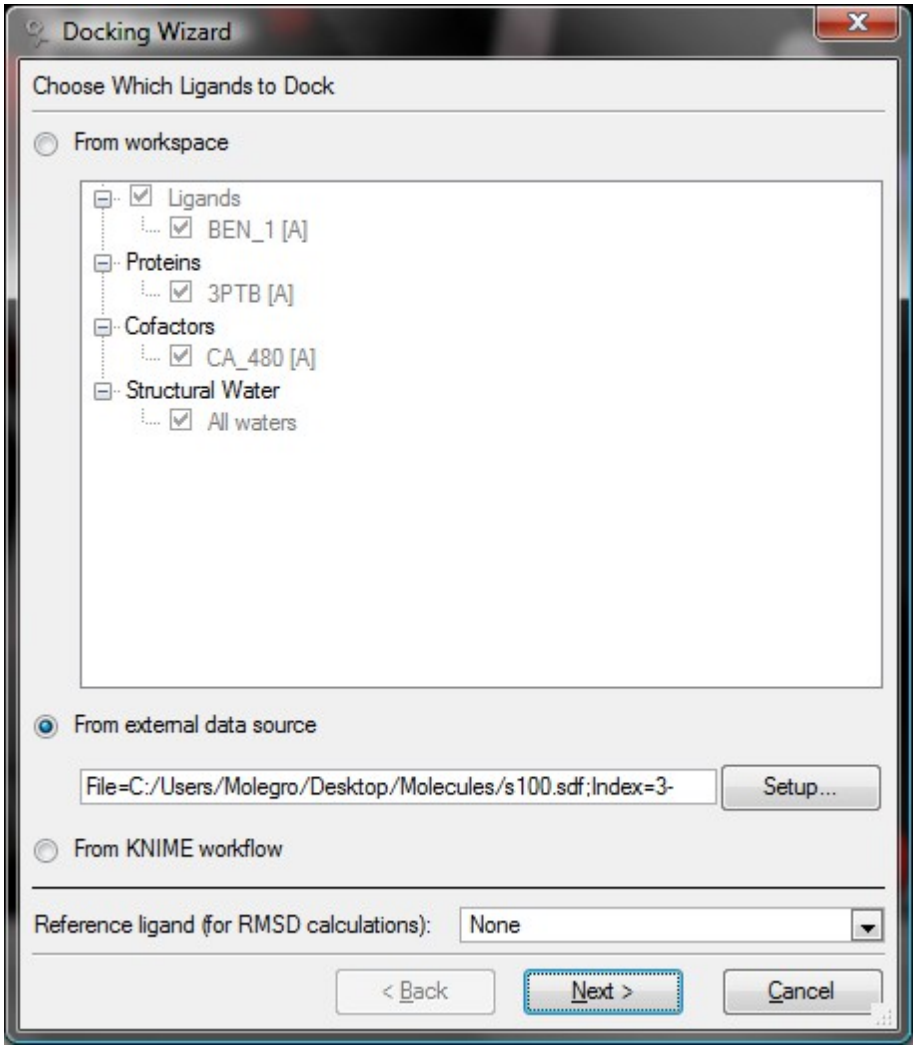


Figure 59: Using data sources from the Docking Wizard.

When choosing **Setup...** a dialog for defining the data source appears:

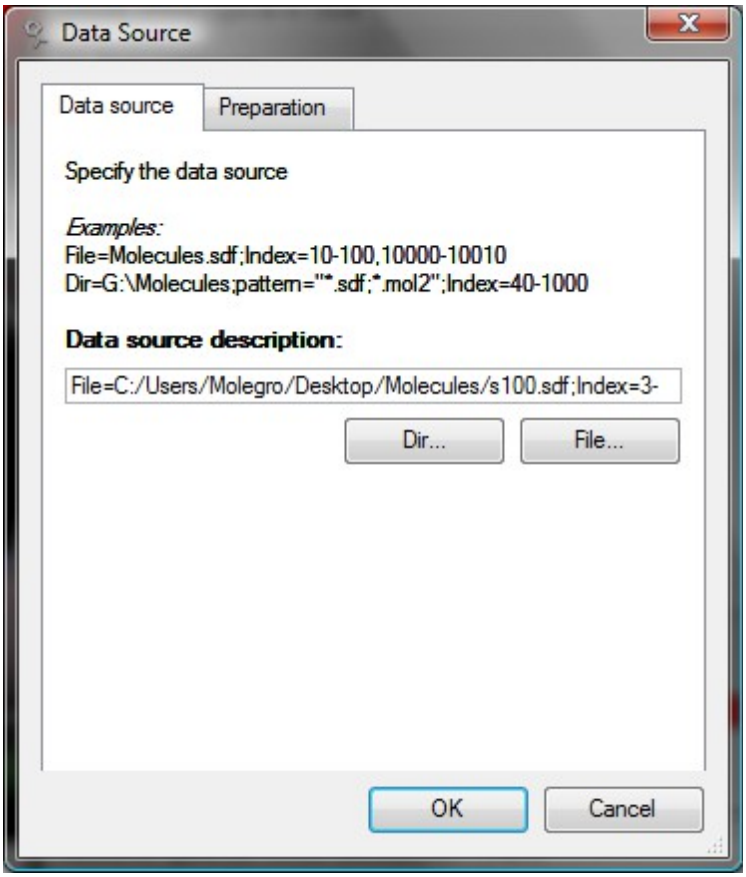


Figure 60: Specifying a data source

Specify the data source on the **Data source description** line input or use either the **Dir...** or **File...** button to choose a directory or file from a dialog.

The **Preparation** tab determines how the data source should be prepared. These settings are described in Section 4.2.

Loading Data Sources Directly into the Workspace

By using the **File | Import From Datasource...** menu item it is possible to directly load a number of molecules into the workspace. This can be useful for importing a small subset of the molecules in a data source to check that the parsing and preparation is okay. Notice that all molecules are loaded into memory which can make the system slow to work with.

The data source wizard that appears is identical to the one described under 'Specifying a data source in the Docking Wizard'.

Using Data Sources from a Script

The Dock command will take a data source as input, if it is surrounded by square brackets:

```
DOCK [File=C:/Molecules/steroids.sdf]
```

or

```
DOCK [Dir="C:/Molecules";Pattern="*.sdf;*.mol2";Index=10-100]
```

Notice that all preparation of the data source will be performed according to the settings defined by any previous PREPARE and PARSERSETTINGS statements in the script. E.g:

```
PARSERSETTINGS breakUnrealisticBonds=false;combineMoleculeFragments=true
PREPARE bonds=ifmissing;bondorders=ifmissing;hydrogens=ifmissing;
charges=always;torsiontrees=always;detectcofactors=false

DOCK [File=C:/Molecules/steroids.sdf]
```

6 Docking Functionality

6.1 Cavity Prediction

Potential binding sites (also referred to as cavities or active sites) can be identified using the built-in cavity detection algorithm (see Appendix IV: Cavity Prediction for details).

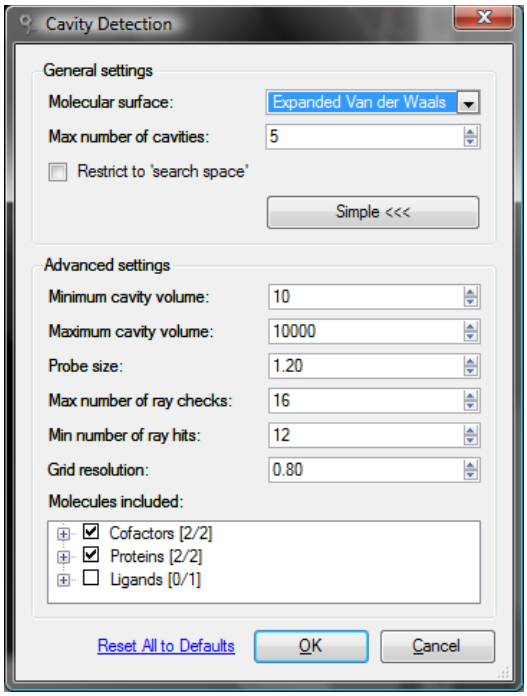


Figure 61: Cavity Prediction dialog.

A dialog is available for detecting cavities (see Figure 61), allowing to customize the sensibility (and type) of cavity search. The **Cavity Prediction** dialog can be invoked via the context menu in the **Workspace Explorer** (in

the **Proteins** category) or by selecting the **Detect Cavities** menu option from the **Preparation** menu.

For large targets a global search for cavities may be slow or result in too many possible binding sites. It is possible to restrict the search to the sphere defined by the current search space (using the **Restrict to 'search space'** checkbox). This makes it possible to define an approximate location of the most likely interaction sites, and then perform the cavity detection within this volume.

The advanced settings are described in Appendix IV: Cavity Prediction. Notice it is possible to choose which molecules should be taken into account when performing the cavity detection.

Cavities found are listed in the **Workspace Explorer** in the **Cavities** category. Visualization of the cavities can be toggled on and off. Moreover, volume and area are listed for each cavity.

Cavities may be deleted from the workspace by selecting them in the workspace explorer and choosing **Remove cavity from workspace**. It is also possible to merge cavities by choosing **Merge With Other Cavity...**

Notice: If no cavities are identified, ligands can only bind to the surface of the protein (or the cavity is too small to be detected). This situation makes it more difficult for the docking engine to identify the correct binding modes.

6.2 Constraints

Constraints are limitations imposed on the molecular system based on chemical insight or knowledge. Constraints can dramatically increase docking accuracy and speed, as they often limit the search space considerably.

There are two fundamental kinds of constraints:

- Hard Constraints: The docking engine tries to fully satisfy these constraints, i.e. a hard constraint could be used to force a specific ligand atom to be in a given region. The docking engine will enforce these constraints by moving or modifying the poses during docking. If several hard constraints exist, and none of them are satisfied, the system will choose to satisfy an arbitrary one. Notice that this means that not necessarily all hard constraints are satisfied. *If a hard constraint is not satisfied, it will add 100 units to the soft constraint energy penalty*.
- Soft Constraints: These act as extra energy terms, and contribute to the overall energy of the system. As such, they can be more or less satisfied. They can for example be used to reward ligands in certain regions. If several (enabled) soft constraints exist in the workspace, they are ALL taken into account (i.e. several extra terms are added to the overall docking energy function while docking).

Creating Constraints.

Distance constraints constrain ligand atoms to a given position in 3D-space (see Figure 62). They are used to constrain some or all atoms of a ligand to the vicinity of this position. Distance constraints are visualized as an inner and outer sphere where some ligand atoms must be present between the spheres.

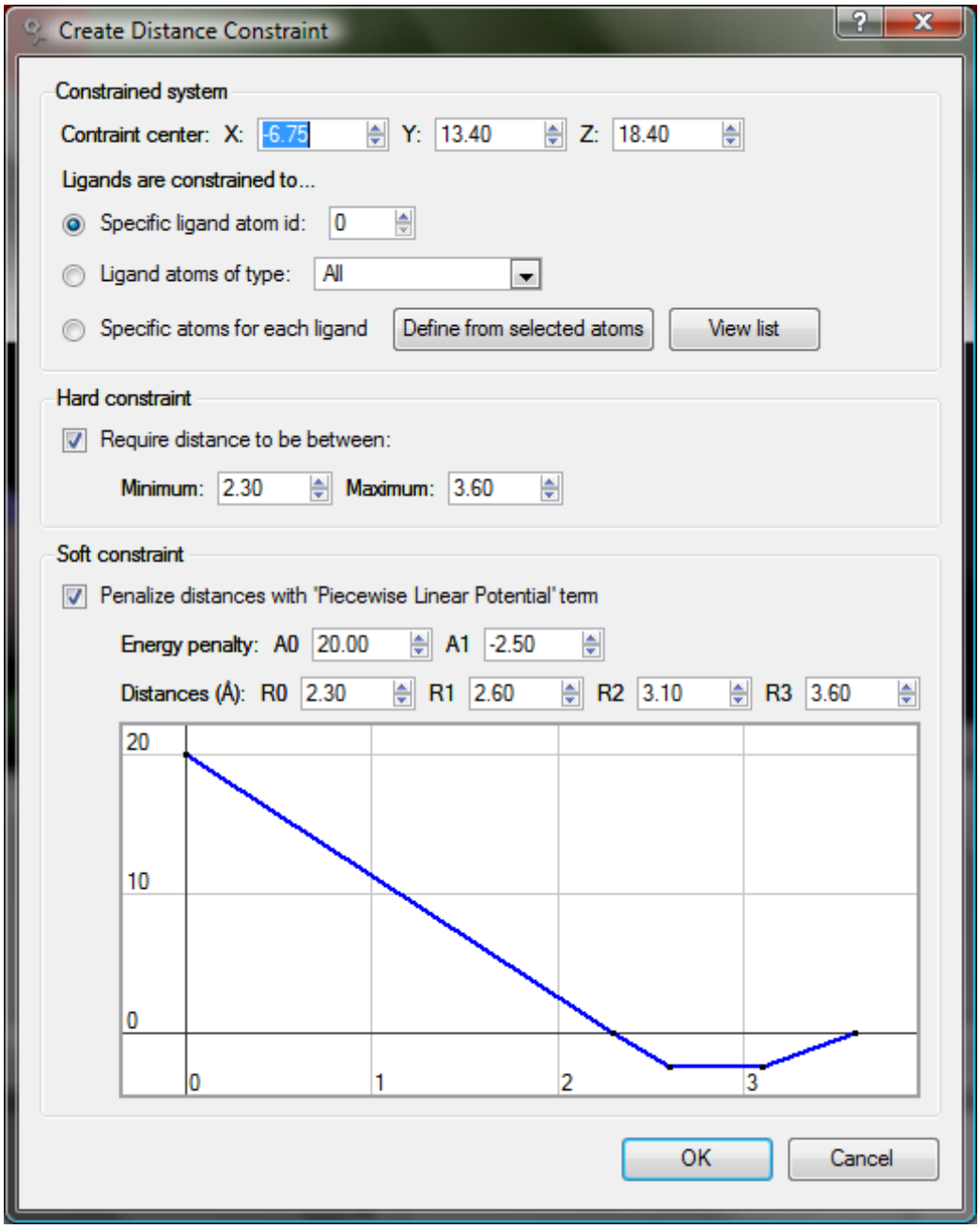


Figure 62: The Distance Constraint dialog.

The **Distance Constraint** dialog can be invoked either via the context menu on an atom (**Create Distance Constraint**), or by selecting one or more atoms, and using the context menu option (**Set Selection as Center of**

Distance Constraint). If several atoms are chosen, their mean position will be set as center for the constraint.

The top panel allows the user to modify the location of the constraint (**Constraint Center**). It also controls which parts of the ligand should be constrained. Either a single atom (the **Specific Ligand Atom (ID)** option) or multiple atoms (the **Ligand Atoms of Type** drop down menu). The different choices for multiple atoms are: **All** (meaning all atoms), **None** (which causes the constraints to try to remove atoms within the constraint range), **Hydrogen Donor**, **Hydrogen Acceptor**, **Hydrogen Donor or Acceptor (Both)**, **Non-polar**, and **Ring Atoms** (atoms in aromatic or aliphatic rings). Additionally, it is also possible to specify ligand atoms from a current selection of atoms using the **Specify atoms for each ligand** option and pressing the **Define from selected atoms** button. This also applies to more than one ligand, which makes it easier to constrain specific atoms for a set of ligands present in the workspace. The **View list** button can be used to inspect the current set of selected ligand atoms.

A **Hard constraint** (see above) range can be specified. If this is enabled the docking engine will try to force the selected ligand atoms to be within this range.

The bottom panel (**Soft constraint**) allows the user to specify a specific potential applied to the selected ligand atoms. The potential is a piece-wise linear potential, which is the same type as used in the docking score function (see Appendix I: MolDock Scoring Function). It is shown graphically in the graph at the bottom.

When applying soft constraints the following procedure is used: for all chosen ligand atoms (as defined by the **Ligands atoms of Type** or **Specific ligand atom** input fields) the distance between the center of the constraint and the atom is calculated. The potential is then evaluated for all these distances, *but only the lowest energy is returned as the soft constraint energy.* That is, only the atom with the lowest energy relative to the constraint potential is taken into account. The reason for this is, that if you for example want to reward ligands with a hydrogen acceptor close to hydrogen donor in the protein, it does not make sense to punish other atoms in the vicinity of the constraint if one hydrogen acceptor is already at its optimal distance from the donor.

Ligand Atom Constraints

Another type of constraint is the **Ligand Atom Constraint**. It is used to constrain a specific ligand. Since **Ligand Atom Constraints** are defined using a list of atom IDs, they are specific to ligands and are only applied to the ligand on which they are defined.

To create a **Ligand Atom Constraint**, select a number of atoms (in the same ligand) in the **Visualization Window**. Ensure that no other objects are

selected, and choose **Constraint Selected Ligand Atoms** from the context menu (right-click mouse button). It is also possible to use the context menu on a single ligand atom (**Create Ligand Atom Constraint**) without performing a selection.

The **Ligand Atom Constraint** dialog will appear (see Figure 63). It is possible to modify the list of atoms in the ligand by entering a comma-separated list of IDs.

Notice: **Ligand Atom Constraints** are always *soft constraints*. It is possible to choose whether the chosen atoms in the ligand should be *rewarded* or *penalized* for contacts with the target molecules (proteins, cofactors and water).

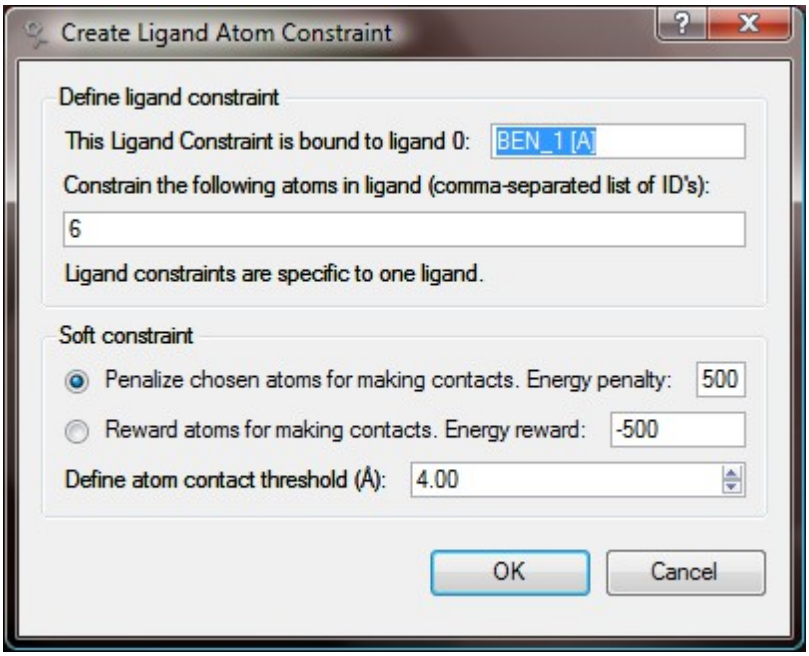


Figure 63: Ligand Atom Constraint dialog.

The criteria for contact used here is purely based on the distance between the chosen ligand atoms and the closest atom in any target molecule. The distance threshold for defining contacts can also be customized (using the **Define atom contact threshold (Å)** input field).

Typical Uses

Constraints are useful if something about the system is known in advance. If perhaps a hydrogen bond from a hydrogen donor was known to be present – a distance constraint could be set up at the position of the protein hydrogen donor, and a hard constraint could force hydrogen acceptors in the ligand to satisfy the hydrogen bond.

6.3 Docking Wizard

When all the molecules have been prepared, the docking can commence. To start the **Docking Wizard**, select **Docking** | **Docking Wizard**. A shortcut is provided by clicking on the docking icon (gear wheel) on the tool bar. Additionally, the keyboard shortcut **F1** is available.

Notice: In order to initiate the docking, at least one protein and one ligand molecule have to be present in the workspace.

Choose Which Ligands to Dock

The first action is to choose which ligands to dock:

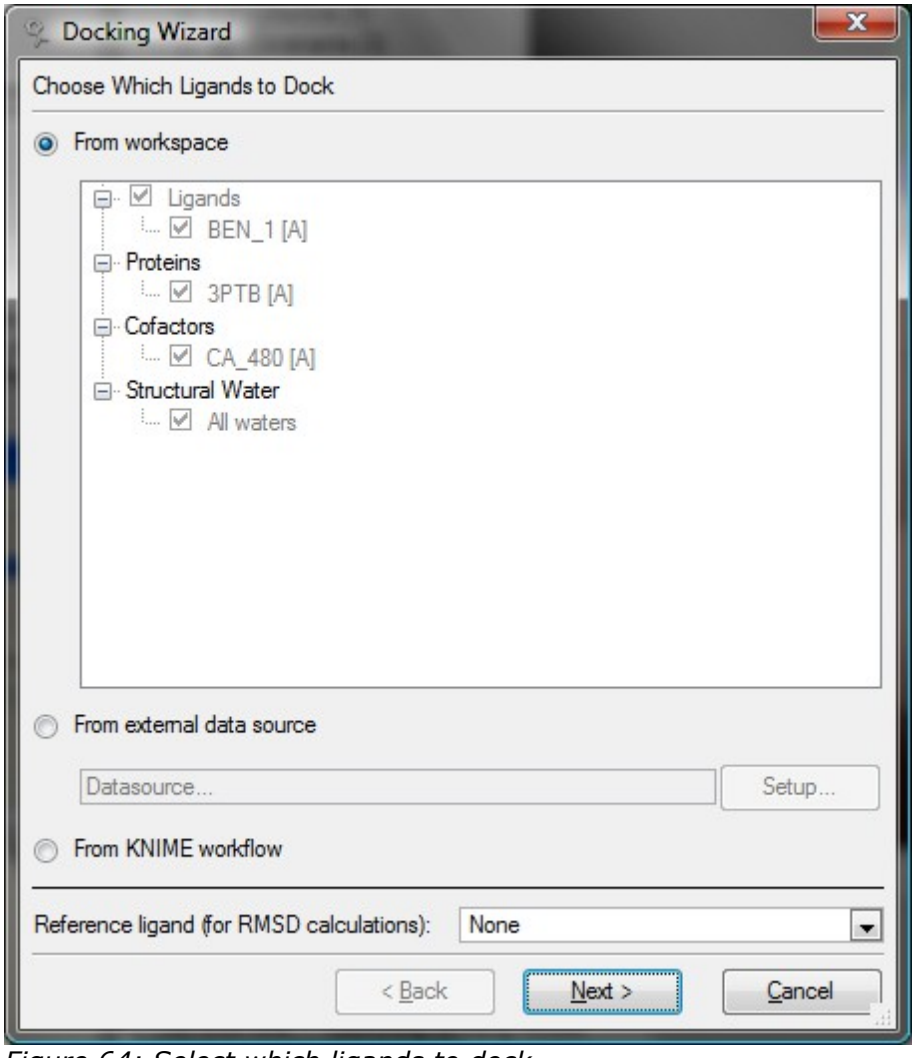


Figure 64: Select which ligands to dock.

If more than one ligand is available in the workspace, the user can select which ones to use by clicking on the corresponding molecules in the window. If more than one ligand is selected, all selected ligands will be docked one at a time. Water molecules and cofactors (if any) are always included in the docking

simulation (remember to remove them from the workspace if they should not be included). Moreover, a reference ligand can be specified at the bottom. The reference ligand is used to calculate the root-mean-squared deviation (RMSD) between the reference ligand and the docked pose. The reference ligand - or ligands - are only available if they are compatible (w.r.t. symmetry, identical number of heavy atoms, etc.) with the ligands selected for docking.

Notice: If more than 10 ligands are present in the workspace, a subset of the ligands can be selected for docking using the **Specify ligand range** option (not shown on Figure 64).

It is also possible to dock ligands from an external data source, see Section 5.2 for details about data sources.

The last option **From KNIME workflow** makes it possible to dock ligands using the KNIME workflow system. See the Molegro KNIME Extensions Installation & usage guide for more details (available from www.molegro.com/knime/).

Choose Scoring Function and Define Binding Site

MVD includes *MolDock Score* [THOMSEN 2006] and PLANTS Score [KORB 2009] for evaluating docking solutions. The *MolDock Score* is further described in Appendix I: MolDock Scoring Function and PLANTS Score is described in Appendix II:PLANTS Scoring Function.

Grid-based versions of the scoring functions are also available. The *MolDock Score [Grid]* is a grid approximation using the same energy terms as the *MolDock Score* except that hydrogen bond directionality is not taken into account. *PLANTS Score [Grid]* is a grid approximation using the same energy terms as the *PLANTS Score*. The grid-based scoring functions provide a 4-5 times speed up by precalculating potential-energy values on an evenly spaced cubic grid (see Appendix XIV: Grid-based Scores for more details).

The following options are available for the MolDock Scoring functions:

The **Ignore distant atoms** option is used to ignore atoms far away from the binding site. Thus, atoms more than **Radius** angstroms away from the center of the binding site are ignored in the scoring function. This reduces the overall computing time significantly when working on large molecules. Notice that charged atoms (capable of long-range interactions) are always taken into account in the scoring function.

The **Enforce hydrogen bond directionality** option is used to check if bonding between potential hydrogen bond donors and acceptors can occur. If hydrogen bonding is possible, the hydrogen bond energy contribution to the docking score is assigned a penalty based on the deviations from the ideal bonding angle. Using this option can significantly reduce the number of unlikely hydrogen bonds reported.

The **Ligand evaluation** can also be customized: **Internal ES** toggles whether internal electrostatic interactions should be calculated for a pose, **Internal Hbond** toggles whether a pose should be allowed to have internal hydrogen bonds, and **Sp2-Sp2 Torsions** determines whether an additional dihedral term should be added for taking Sp2-Sp2 bonds into account (see Appendix I: MolDock Scoring Function).

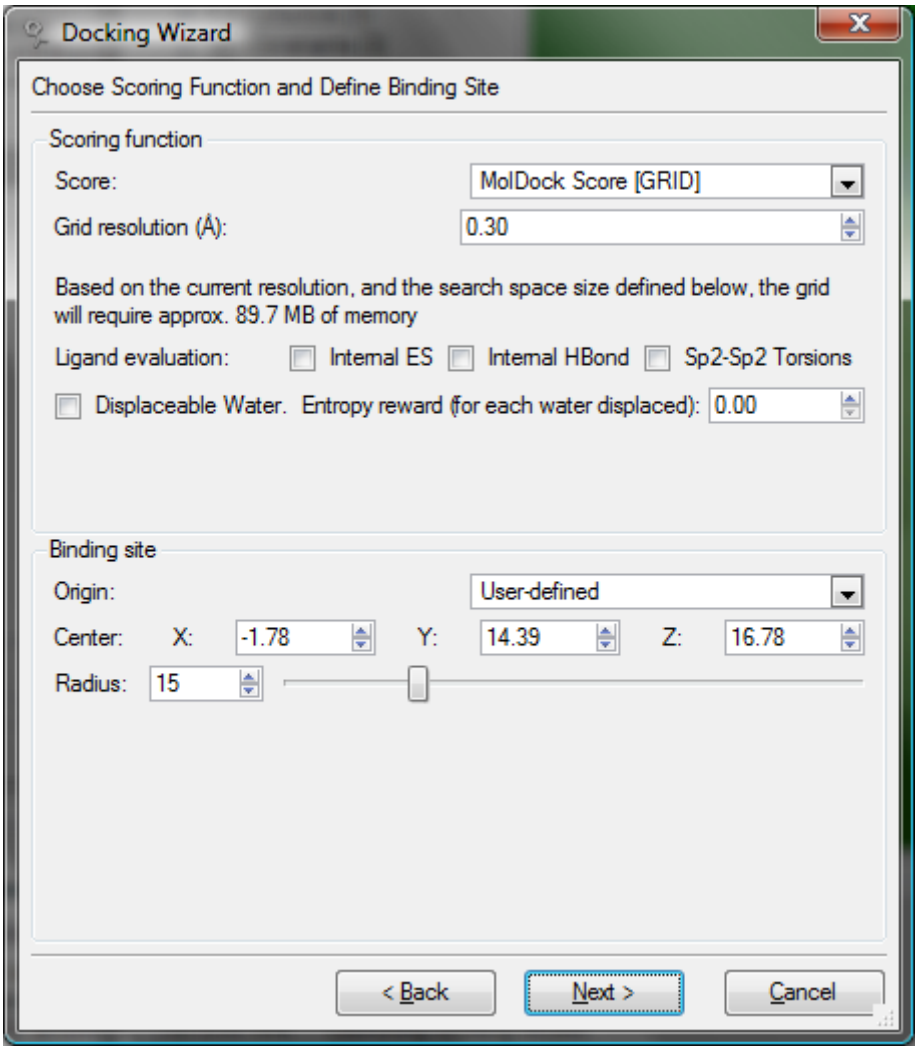


Figure 65: Choosing scoring function.

For the PLANTS Score, the following options are available:

Include hydrogens in torsion term toggles whether or not hydrogens should be included when calculating the Tripos torsion potential (see Appendix II:PLANTS Scoring Function for details about the PLANTS scoring function).

The **Ignore distant atoms** option is used to ignore atoms some distance away from the binding site and is similar to the option for the MolDock score.

If water molecules are available in the workspace, it is possible to include

displaceable water evaluation by enabling the **Displaceable Water** option (see Section 9.1 for more details).

For the grid-based scoring functions, the **Grid resolution** option (not shown in Figure 65) can be used to set the granularity of the generated energy grids.

The **Binding site** specifies the region of interest and thus where the docking procedure will look for promising poses (ligand conformations). The **Origin** determines which area of the protein is expected to include the binding site.

If cavities have been identified the user can pick one of these as the preferred area of interest. Further, if a reference ligand is being used, the center of the reference ligand can be used. By default (if no cavities have been identified and no reference ligand is specified), the center of the bounding box spanning all protein(s) will be used. The actual center of the binding site used is listed in the **X**, **Y**, and **Z** boxes in the window. Besides the center of the binding site, a **Radius** can be specified (default is 15 angstrom). The **Search Space** region will be shown in the **Workspace Explorer** in the **Constraints** category.

Notice: A sphere in the **Visualizer Window** indicates the position and size of the current search space region (see Figure 66).

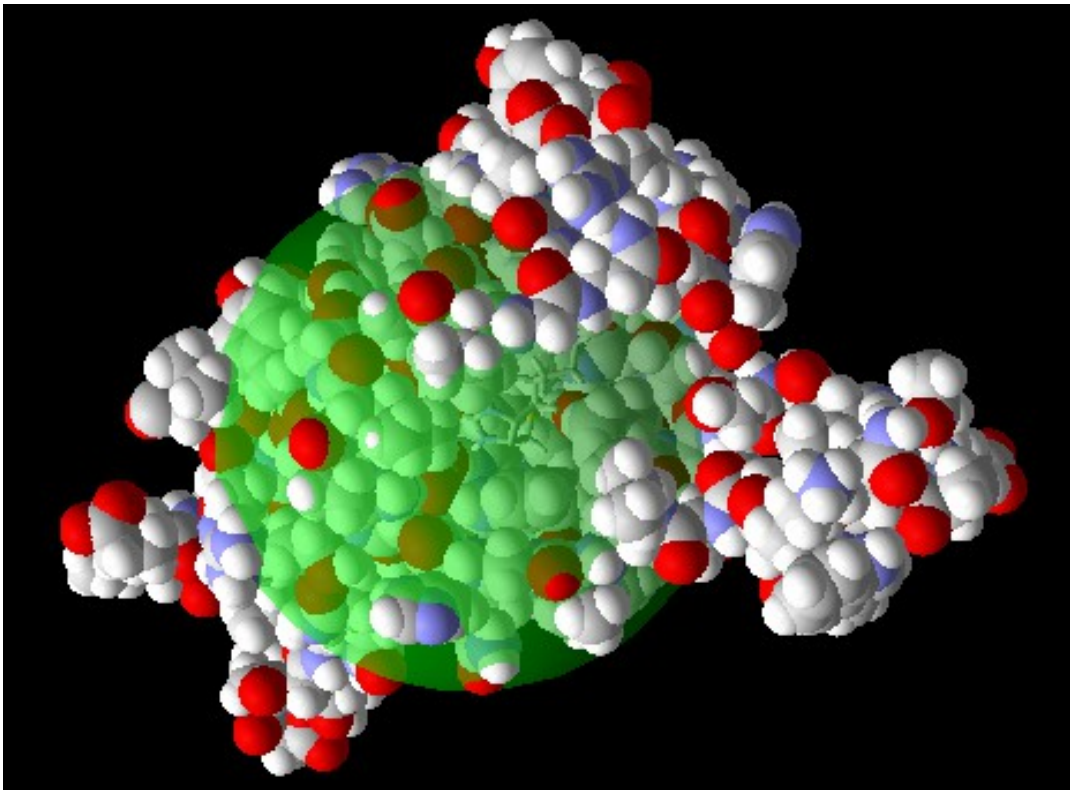


Figure 66: Example of search space region (green sphere).

Enable or Disable Additional Constraints

If constraints (besides the search space region) have been added to the workspace, they can be toggled on and off in the **Enable or Disable Additional Constraints** tab (see Figure 67). In order for a constraint to be meaningful it must be defined within the current search space region.

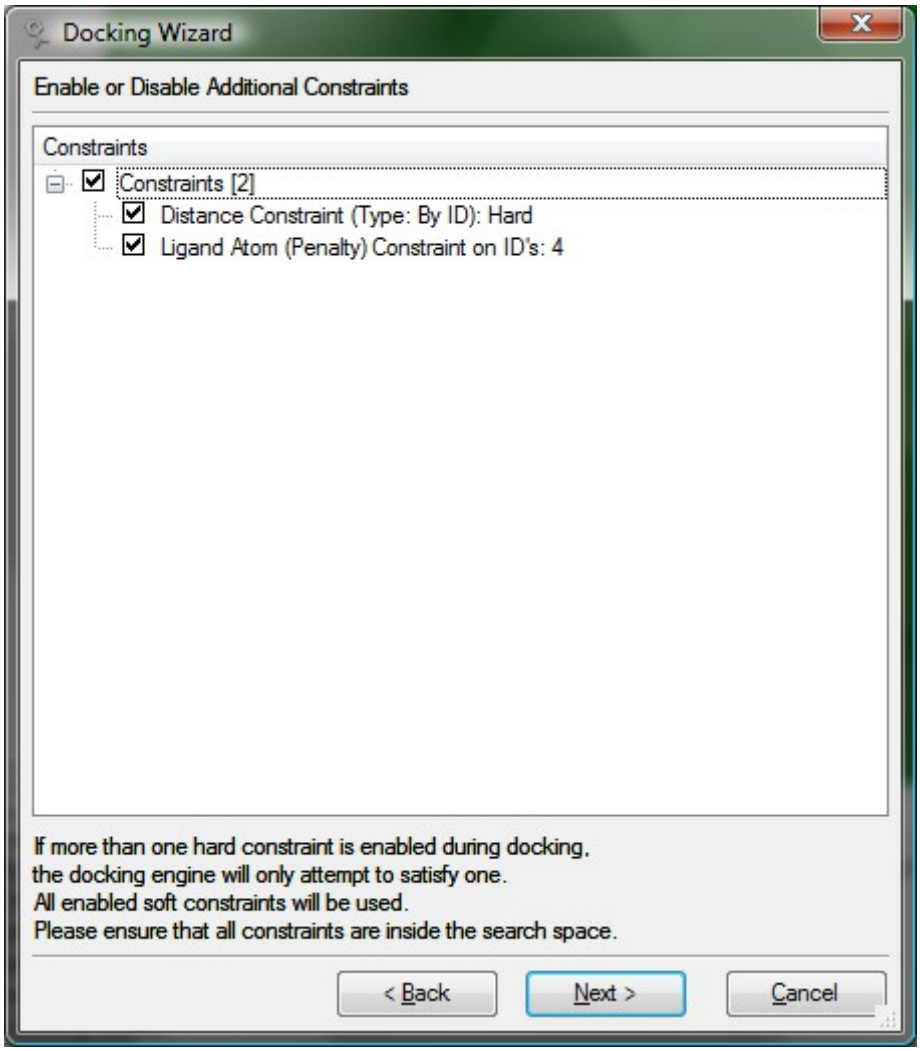

Docking Wizard
Enable or Disable Additional Constraints
Constraints
Constraints [2]
Distance Constraint (Type: By ID): Hard
Ligand Atom (Penalty) Constraint on ID's: 4
If more than one hard constraint is enabled during docking,
the docking engine will only attempt to satisfy one.
All enabled soft constraints will be used.
Please ensure that all constraints are inside the search space.
< Back
Next >
Cancel

Figure 67: Enabling or disabling user-defined constraints.

Choose Search Algorithm

MVD includes three search algorithms for molecular docking, *MolDock Optimizer* [THOMSEN 2006], *MolDock SE* (simplex evolution), and I*terated Simplex*.

The **Number of runs** specifies the number of times that the docking simulation is repeated for each ligand chosen to be docked. Sometimes more than one run is needed to identify promising poses (in particular for ligands having more than 15 flexible torsions or if no promising cavities exist). If cavities have been identified (see Section 6.1) the poses found by the search

algorithm can be constrained to the region spanned by the cavity (by using the **Constrain poses to cavity** option). This option greatly reduces the overall docking process and increases the accuracy of the docking procedure. However, if the ligand does not bind in the region specified by the selected cavity, this option should be disabled.

The **After Docking** settings make it possible to perform two post-docking steps: **Energy Minimization** performs a short Nelder-Mead Simplex minimization of the translation, orientation and flexible dihedral angles of the found poses using the MolDock scoring function. This step can be used to slightly refine the docking results – the evaluation is always performed with a non-grid version of the MolDock scoring function, thus preventing any inaccuracies due to energy grid approximations. Also the non-grid MolDock scoring function is able to more precisely take hydrogen bonding geometries into account. Enabling this option will usually result in very tiny improvements, and the option is disabled by default. **Optimize H-Bonds** optimizes the position of the hydrogens for any hydrogen donors (both in the Ligand and in the Proteins). The default behavior for the MolDock score is to only evaluate hydrogen bond angle geometry for hydrogen bonds where the hydrogen positions are fixed (non-rotatable). By optimizing both protein and ligand positions first, additional geometric constraints can be used to evaluate the quality of a hydrogen bond. Notice that enabling this will always slightly *raise* the energy (since the geometric hydrogen bond terms are penalties imposed on the hydrogen bond energies) – this does not mean that the solution is worse after optimizing the hydrogen bonds, but rather that the more accurate evaluation has made it possible to impose additional penalties on the hydrogen bonding geometry. By default h-bond optimization is enabled. Notice that the **After Docking** settings become unavailable if the workspace contains docking templates, sidechain flexibility descriptors, or constraints.

Notice: For large ligands with more than 10-15 flexible bonds, 20-50 runs are sometimes needed. Using the MolDock SE search algorithm and the grid-based version of the docking scoring function can reduce the computational load significantly (good results have been reported using this combination and setting the **Number of runs** to 50).

The **Parameter Settings** show the parameters used by the MolDock Optimizer search algorithm. The default values shown are generally suitable for most docking tasks. See Appendix III: MolDock Optimizer for details on MolDock Optimizer parameter settings.

The MolDock SE and Iterated Simplex algorithms (and their parameters) are further described in Appendix XII: MolDock SE and Appendix XIII: Iterated Simplex.

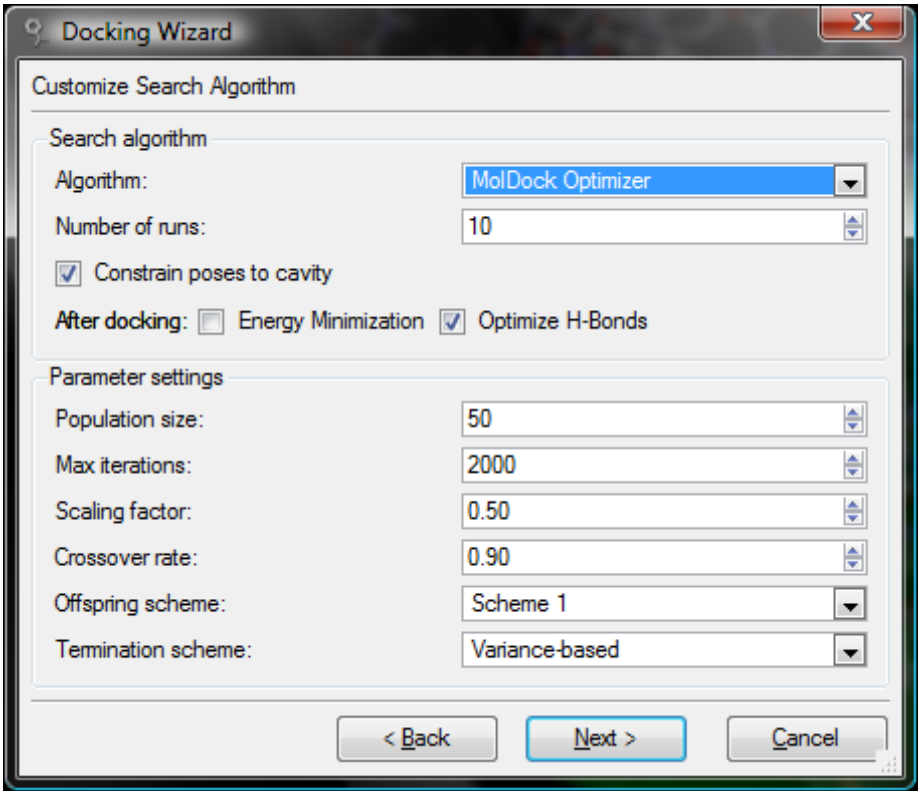


Figure 68: Customizing the search algorithm.

Pose Clustering

Instead of returning only one final pose for each docking run, it is possible to return multiple poses representing different potential binding modes. This can be useful when the best-scoring (i.e. lowest-energy) pose does not represent the native binding mode or when multiple binding modes exists.

Here, clustering can be used to reduce the number of poses found during the docking run and only the most promising ones will be reported.

If multiple poses are returned for each run, the following options are available:

- Limit the number of poses reported (**Max number of poses returned**).
- Only report poses with energies less than a user-defined threshold (**Enable energy threshold**).
- Cluster poses using the specified RMSD threshold (**Cluster similar poses**). Poses found during the docking run will be clustered (put into bins) using the RMSD criteria. See Appendix V: Clustering Algorithm for a detailed description of the clustering algorithm used. Only the lowest-energy representative from each cluster will be returned when the docking run is completed. Increasing the RMSD threshold will increase the diversity (with respect to RMSD) of the poses returned.
- The **Ignore similar poses** option is used to avoid reporting to similar

poses when conducting multiple runs (docking the same ligand). All poses returned from the runs will be clustered and similar poses are removed (keeping the best-scoring one). Depending on the RMSD threshold specified, more or less diverse poses (combined for all the runs) will be reported.

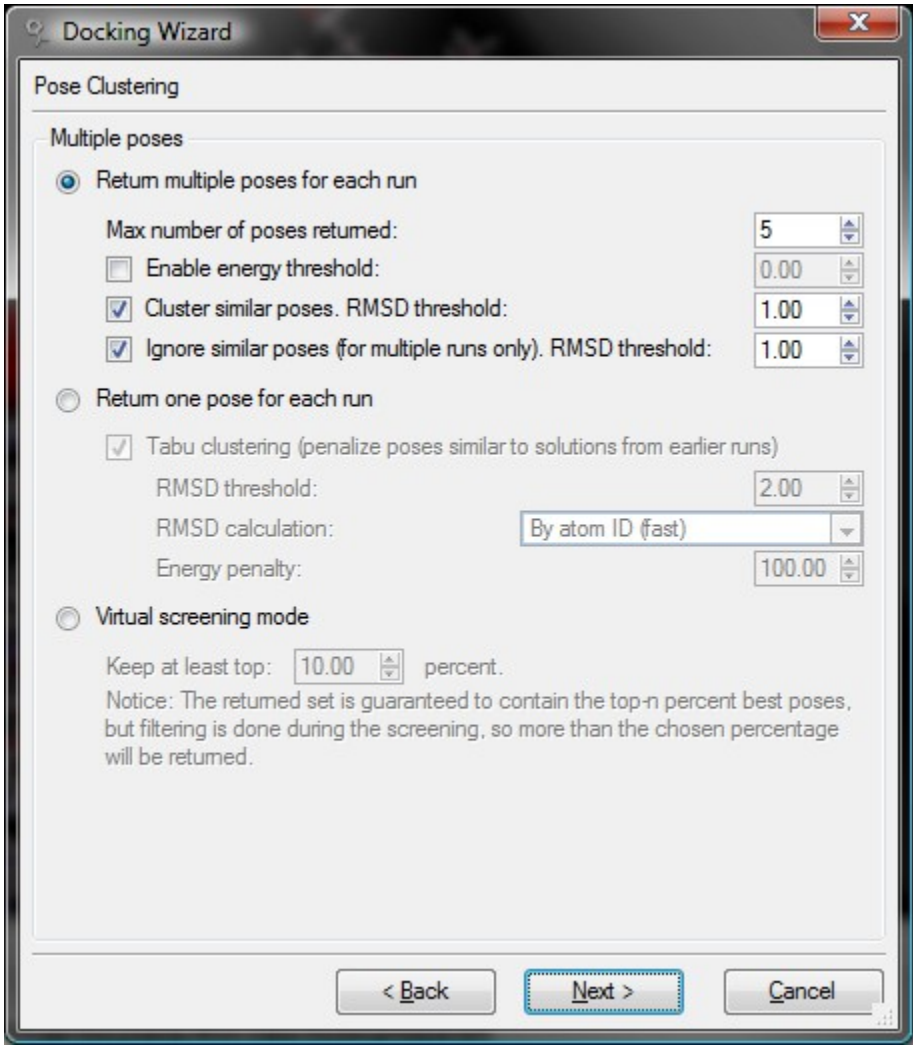

Docking Wizard
Pose Clustering
Multiple poses
Return multiple poses for each run
Max number of poses returned:
5
Enable energy threshold:
0.00
Cluster similar poses. RMSD threshold:
1.00
Ignore similar poses (for multiple runs only). RMSD threshold:
1.00
Return one pose for each run
Tabu clustering (penalize poses similar to solutions from earlier runs)
RMSD threshold:
2.00
RMSD calculation:
By atom ID (fast)
Energy penalty:
100.00
Virtual screening mode
Keep at least top:
10.00
percent.
Notice: The returned set is guaranteed to contain the top-n percent best poses, but filtering is done during the screening, so more than the chosen percentage will be returned.
< Back
Next >
Cancel

Figure 69: Pose clustering options.

Notice: The actual number of poses returned may be lower than the maximum number of poses specified in **Max number of poses returned**. For instance, the energy- or clustering-threshold options can reduce the number of poses returned (if poses have higher energies or are too similar). However, the overall best-scoring pose will always be returned.

If only one pose is returned per run (**Return one pose for each run**), a special clustering technique (called 'Tabu Clustering') can be applied.

When using this clustering technique each found solution is added to a 'tabu list': during the docking simulation the poses are compared to the ligands in this 'tabu list'. If the pose being docked is closer to one of the ligands in the

list than specified by the **RMSD treshold**, an extra penalty term (the **Energy penalty**) is added to the scoring function. This ensures a greater diversity of the returned solutions since the docking engine will focus its search on poses different from earlier poses found. It is possible to specify whether RMSD calculations should be performed by comparing atom ID (which is the fastest choice and the default choice) or if intrinsic ligand symmetries should be taken into account (which is slower but more accurate). Tabu clustering is performed per ligand – when a new ligand is docked the tabu list is cleared. Notice that the tabu list gets longer for each run – so when docking many runs for each ligand tabu clustering may slow the system.

For virtual screening runs, the **Virtual screening mode** option is particularly suitable for returning a percentage of the top-ranked compounds found during the virtual screening run (the percentage can be specified in the dialog). Since the set of top-ranked poses found are updated dynamically during the virtual screening run more than the specified percentage of poses will be returned. Notice: setting the percentage to 0 will toggle off pose clustering and the best-scoring pose for all ligands will be returned.

Errors and Warnings

The **Docking Wizard** reports errors and warnings found, such as non-bonding atoms in molecules, steric clashes between atoms, unsupported residues, missing hydrogens in proteins, etc. A detailed description of each warning and error is shown at the bottom of the **Errors and Warnings** tab (see Figure 70).

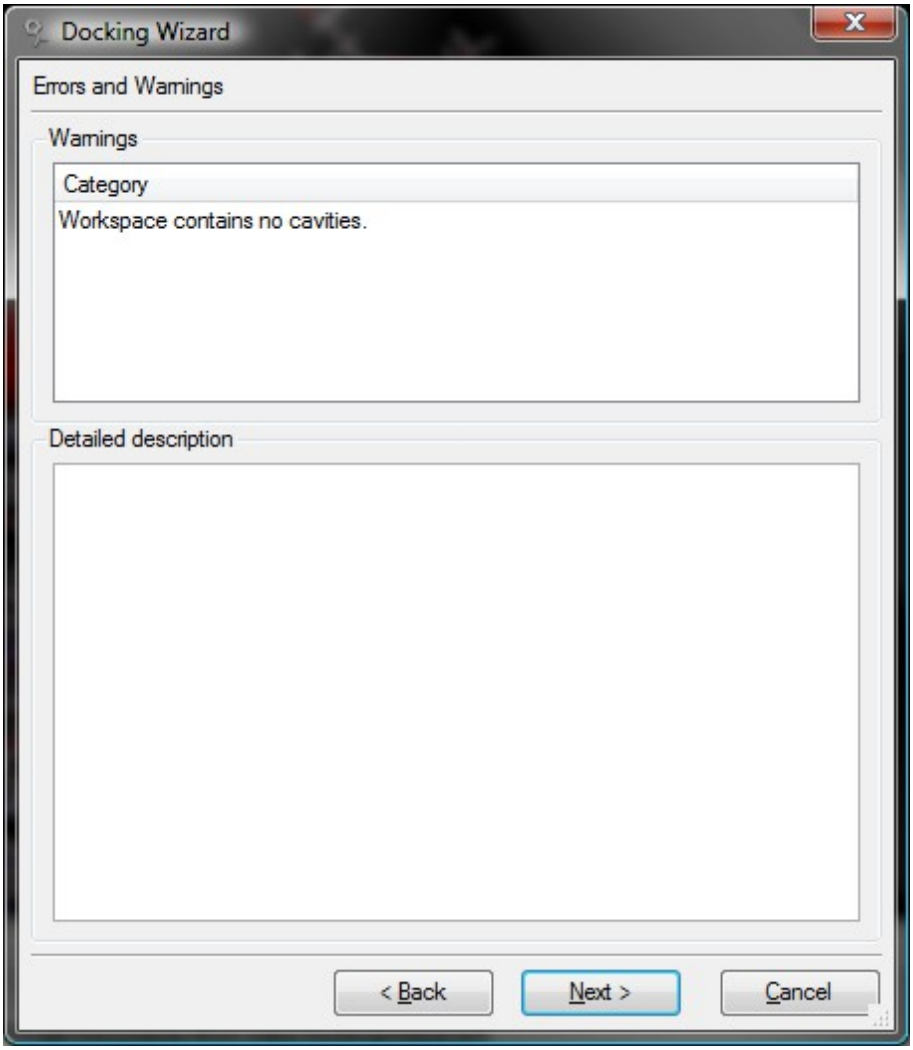


Figure 70: Any warnings or errors are shown on the last page in the wizard.

Setup Docking Execution

In the final tab (see Figure 71), three choices are available for executing the docking simulation. **Run docking in separate process** is the default choice, which creates a MVD script that is executed in an external process (Chapter 17 describes the MVD Scripting Interface in more details). A copy of the current workspace is used, so the user can continue working with the current workspace without interfering with the docking simulation (e.g. add/remove molecules, change preparation, etc.). The second choice **Create a docking script job, but do not run it now** creates a docking script using the currently selected parameter settings. The generated script is saved in the directory specified in the **Output directory** (see below) and can be used to start up the docking simulation on other computers. The third choice, **Run docking in multiple processes**, makes it possible to use multiple cores or GPUs during the docking simulation (see Chapter 6.4 for more information on docking with multiple processes). The fourth choice **Start job on Virtual Grid** creates a

grid job and spawns the Virtual Grid Controller (see Chapter 15 for more details about Virtual Grid execution).

When enabling the **Edit script manually** option, a tab page containing the MVD script is shown making it possible to manually edit the script before starting the docking run.

The **Output directory** specifies where the docking data (MVD script file, MVD script log file, docking results file, and found poses) will be stored. The MVD script file (`script.mvdscript`) contains the scripting commands automatically generated to perform the docking simulation. The MVD script log file (`ScriptLog_timestamp.txt`) is a time-stamped log file containing log information generated by the script interpreter. Details about the poses returned after the docking simulation (e.g. docking score, affinity, specific energy terms, and pose Mol2 filename) are included in a `mvdresults` file (`DockingResults.mvdresults`). The `mvdresults` file is used by the **Pose Organizer** to show detailed information about the poses and to dynamically load the molecular structure of the poses (see Section 7.1 for more details). Each pose is stored in either Mol2 or MVDML format (as chosen in the **Save found poses as** combobox). The poses are used by the **Pose Organize**r to show the 3D conformations of the poses in the **Visualization Window**.

The **Pose name pattern** specifies how the poses should be named when saved. By default '[$ID] $Name' is used but the pattern can be changed if white spaces or square brackets should be omitted from the pose filename.

If the **Create SMILES in MVDResults file** is enabled, a column with SMILES strings is added to the MVDResults output. This makes it possible to visualize 2D depictions of the molecules in Molegro Data Modeller when analyzing docking data.

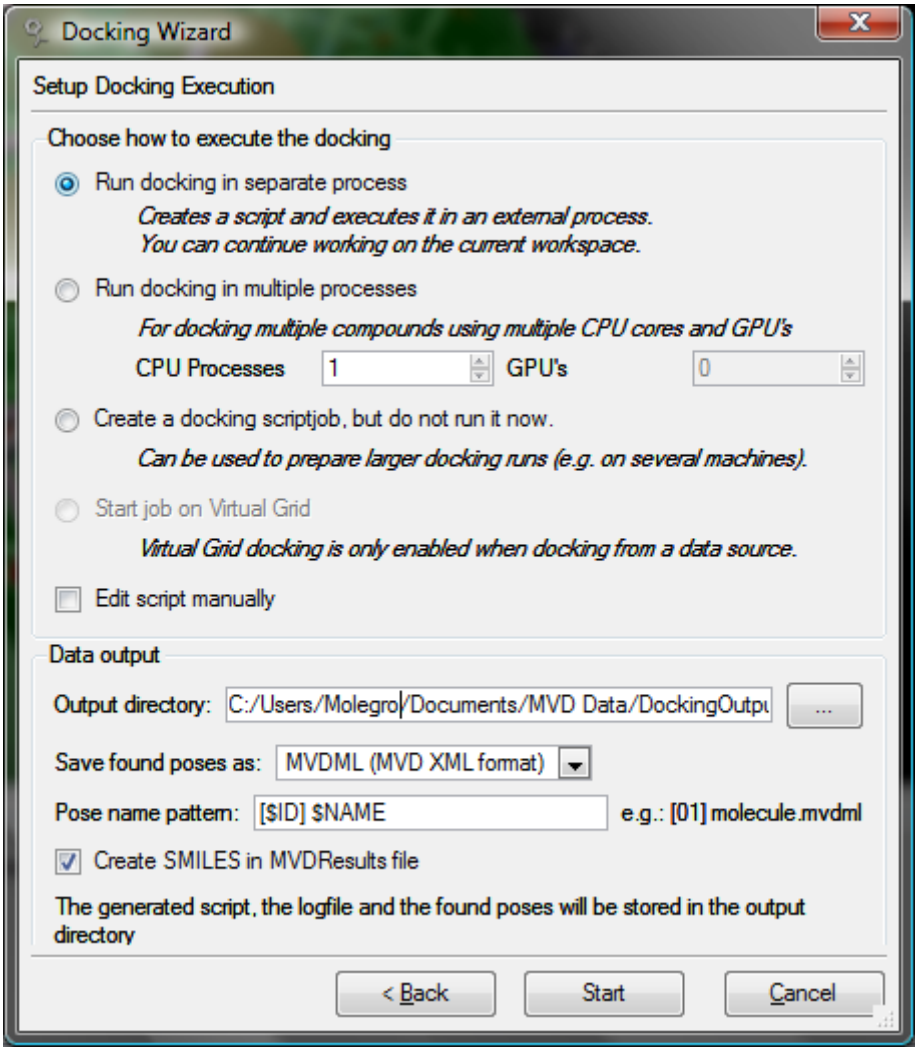

Docking Wizard
Setup Docking Execution
Choose how to execute the docking
Run docking in separate process
Creates a script and executes it in an external process.
You can continue working on the current workspace.
Run docking in multiple processes
For docking multiple compounds using multiple CPU cores and GPU's
CPU Processes
1
GPU's
0
Create a docking scriptjob, but do not run it now.
Can be used to prepare larger docking runs (e.g. on several machines).
Start job on Virtual Grid
Virtual Grid docking is only enabled when docking from a data source.
Edit script manually
Data output
Output directory:
C:/Users/Molegro/Documents/MVD Data/DockingOutpu
...
Save found poses as:
MVDML (MVD XML format)
Pose name pattern:
[$ID] $NAME
e.g.: [01] molecule.mvdml
Create SMILES in MVDResults file
The generated script, the logfile and the found poses will be stored in the output directory
< Back
Start
Cancel

Figure 71:Setup docking execution.

Finally, when the **Start** button is pressed, the docking run will start and the **Molegro Virtual Docker Batchjob** dialog will pop up - showing the current status and progress of the docking (see Figures 72 and 73).

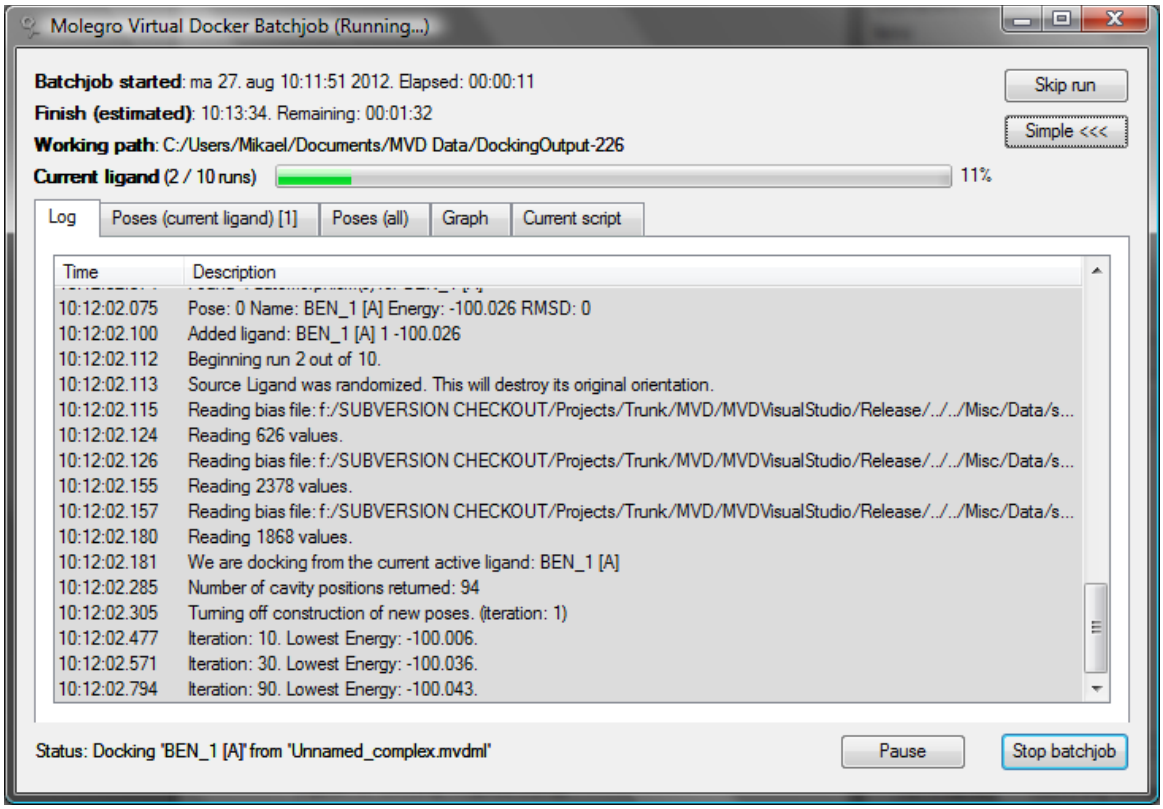

Molegro Virtual Docker Batchjob (Running...)
Batchjob started: ma 27. aug 10:11:51 2012. Elapsed: 00:00:11
Finish (estimated): 10:13:34. Remaining: 00:01:32
Working path: C:/Users/Mikael/Documents/MVD Data/DockingOutput-226
Current ligand (2 / 10 runs)
11%
Skip run
Simple <<<
Log
Poses (current ligand) [1]
Poses (all)
Graph
Current script
Time
Description
10:12:02.075 Pose: 0 Name: BEN_1 [A] Energy: -100.026 RMSD: 0
10:12:02.100 Added ligand: BEN_1 [A] 1 -100.026
10:12:02.112 Beginning run 2 out of 10.
10:12:02.113 Source Ligand was randomized. This will destroy its original orientation.
10:12:02.115 Reading bias file: f:/SUBVERSION CHECKOUT/Projects/Trunk/MVD/MVDVisualStudio/Release/../../Misc/Data/s...
10:12:02.124 Reading 626 values.
10:12:02.126 Reading bias file: f:/SUBVERSION CHECKOUT/Projects/Trunk/MVD/MVDVisualStudio/Release/../../Misc/Data/s...
10:12:02.155 Reading 2378 values.
10:12:02.157 Reading bias file: f:/SUBVERSION CHECKOUT/Projects/Trunk/MVD/MVDVisualStudio/Release/../../Misc/Data/s...
10:12:02.180 Reading 1868 values.
10:12:02.181 We are docking from the current active ligand: BEN_1 [A]
10:12:02.285 Number of cavity positions returned: 94
10:12:02.305 Turning off construction of new poses. (iteration: 1)
10:12:02.477 Iteration: 10. Lowest Energy: -100.006.
10:12:02.571 Iteration: 30. Lowest Energy: -100.036.
10:12:02.794 Iteration: 90. Lowest Energy: -100.043.
Status: Docking 'BEN_1 [A]' from 'Unnamed_complex.mvdml'
Pause
Stop batchjob

Figure 72: Docking Progress dialog.

The **Graph** tab shows the convergence of the population of candidate solutions. The blue graph shows the energy of the best pose and the red graph shows the mean energy of the entire population of candidate solutions (see Appendix III: MolDock Optimizer for more details about the docking simulation and the population terminology). Notice: The red graph is only shown when using the MolDock Optimizer docking algorithm.

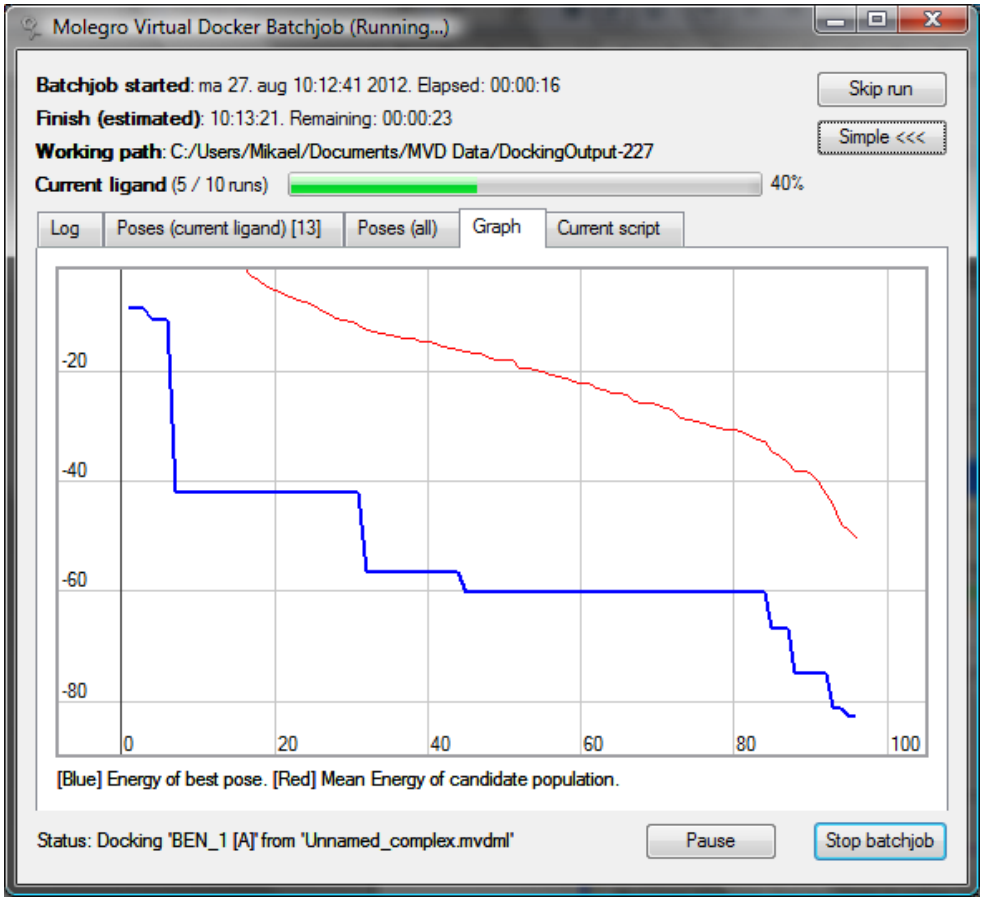


Figure 73: Docking progress dialog with convergence graph shown.

6.4 Run Docking in Multiple Processes

The default docking option, **Run docking in separate process**, is single-threaded, which means only a single CPU core is utilized. By running the docking simulation in multiple processes, it is possible to take advantage of the multiple cores present in modern CPU's or multi-processor systems. In most cases the optimal number of processes correspond to the number of physical (not virtual) cores in the system.

Notice, that there may also be a speed advantage to using multiple CPU threads to feed one or more GPU's, when doing GPU docking (this is mainly the case for fast GPU's). It is even possible to use multiple GPU's (check the number of available GPU's in **Help** | **Show Available CUDA Devices**). The best way to find the optimal CPU/GPU setting is by trying out different configurations on a sample test set of compounds, but usually the best setting is to match the number of CPU processes with the number of physical cores, and the number of GPU's with the available GPU's.

When docking in multiple processes, a different progress dialog is displayed:

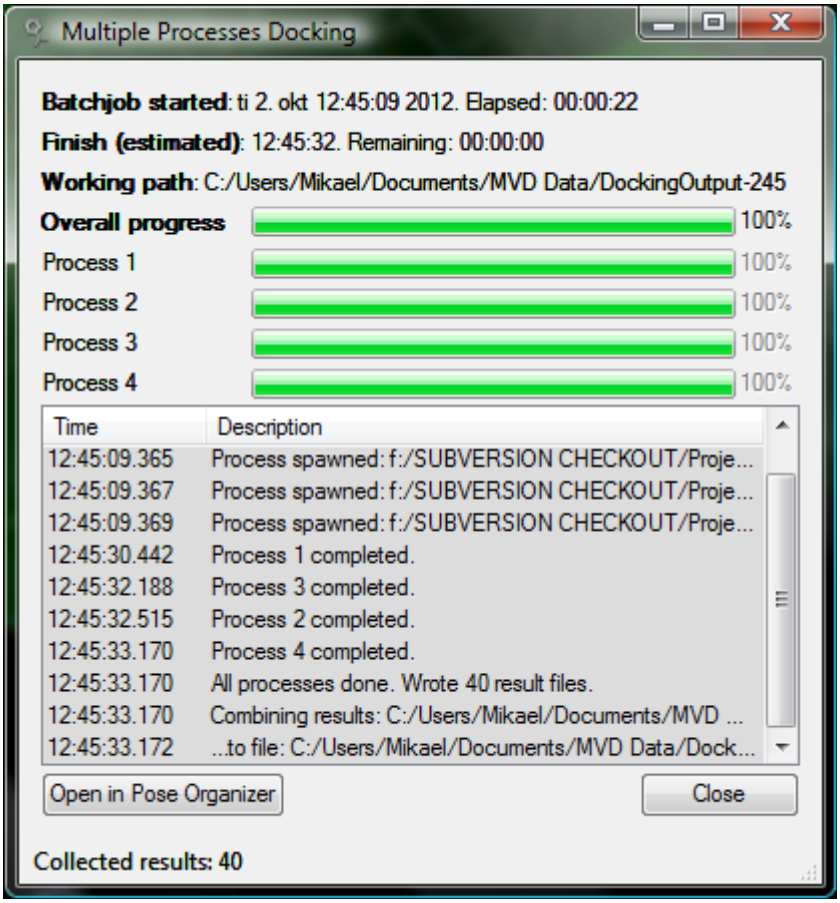

Multiple Processes Docking
Batchjob started: ti 2. okt 12:45:09 2012. Elapsed: 00:00:22
Finish (estimated): 12:45:32. Remaining: 00:00:00
Working path: C:/Users/Mikael/Documents/MVD Data/DockingOutput-245
Overall progress 100%
Process 1 100%
Process 2 100%
Process 3 100%
Process 4 100%
Time Description
12:45:09.365 Process spawned: f:/SUBVERSION CHECKOUT/Proje...
12:45:09.367 Process spawned: f:/SUBVERSION CHECKOUT/Proje...
12:45:09.369 Process spawned: f:/SUBVERSION CHECKOUT/Proje...
12:45:30.442 Process 1 completed.
12:45:32.188 Process 3 completed.
12:45:32.515 Process 2 completed.
12:45:33.170 Process 4 completed.
12:45:33.170 All processes done. Wrote 40 result files.
12:45:33.170 Combining results: C:/Users/Mikael/Documents/MVD ...
12:45:33.172 ...to file: C:/Users/Mikael/Documents/MVD Data/Dock...
Open in Pose Organizer
Close
Collected results: 40

Figure 74: The Multiple Processes Docking Progress Dialog.

In the **Multiple Process Docking Progress Dialog** each thread show its current progress. While running in multiple process docking mode, the main GUI is locked, and cannot be used (this is in contrast to separate process docking, where it is possible to keep using the main GUI). After all processes have finished, it is possible to open the docking results in the **Pose Organizer**, by pressing the **Open in Pose Organizer** button.

There are also other ways to setup multithread docking: by running a Molegro Virtual Grid Client and Server on the same machine, it is also possible to use multiple cores (see Chapter 15.8).

Multiple Process Docking can also be invoked from the command-line, by specifying a '.mvdscript' file, and the number of processes and gpu, using the syntax: -ps <x> and -gpus <x>, e.g.:

```
mvd test.mvdscript -ps 2 -gpus 2
```

6.5 GPU Screening

It is possible to perform screening runs on a Graphics Processor Unit, a GPU, in Molegro Virtual Docker. The advantage of using a GPU is the speed - a modern CPU processor, such as the Intel Core i7-980 XE may deliver around 200 GFLOPS (FLOPS = Floating Point Operations per Second) of computational power, while a modern GPU, such as the GeForce® GTX® 580, may deliver around 1600 GFLOPS. However, GPU's are less flexible than conventional desktop CPU's, which means the software must target GPU's specifically in order to utilize their computational power.

Molegro Virtual Docker offers a special mode for doing docking calculations using the GPU. Since the algorithms had to be adapted to GPU, the approach for GPU Screening is slightly different than the other algorithms in MVD. The algorithm is described in 'The GPU Screening Algorithm'.

Hardware requirements

The implementation in Molegro Virtual Docker is done using Nvidia's CUDA platform for GPU programming. This means a CUDA-capable graphics card from Nvidia is required to use the GPU screening modes. There is no direct requirement on the computational power for the graphics card, but it should be able to deliver more than 100 GFLOPS of computational power, in order to make it preferable to use GPU's instead of CPU's.

Notice that our GPU implementation does not rely on double precision floating point support on the GPU. This means that it is not necessary to use the Nvidia Tesla series hardware for doing scientific calculations. The GPU screening can be performed on standard graphics hardware, such as the Nvidia's GeForce or Quadro series of graphics cards.

Selecting a CUDA device

Some machines may have more than one CUDA device installed: for instance, two graphics cards, or a primary graphics card and a Nvidia Tesla card for scientific computations. One instance of Molegro Virtual Docker will only be able to use a single card at a time.

If you have more than two CUDA devices installed, it is necessary to choose the required device. The default device may be selected from the preferences in MVD. Select **Edit** | **Preferences...** and on the **General** tab, choose the number of the required default CUDA device (a list of the detected CUDA devices and their corresponding IDs may be found by choosing **Help** | **Show Available CUDA Devices**).

The default device, specified in the program preferences, is used unless anything else is specified. It is possible to override the default device settings by either:

- Specifying a CUDA device as a command line parameter when starting Molegro Virtual Docker. This is done using the 'cudadevice' parameter, e.g.: *MVD.exe -cudadevice 1*
  where the number refers to the indices as listed above.
- In a mvdscript, specify the desired device using the 'CUDADEVICE' command, e.g. in a script, add the following line (before the dock and optimizer command):

```
CUDADEVICE 1
OPTIMIZER populationsize=50;cavity=true;...
LOAD "Unnamed_complex.mvdml"
DOCK
```

Setting up a GPU Screening run

In order to set up a GPU Screening run, prepare the workspace as for a normal docking simulation and start the Docking Wizard. Now go to the **Customize Search Algorithm** and set the algorithm to **GPU Screening (CUDA)**.

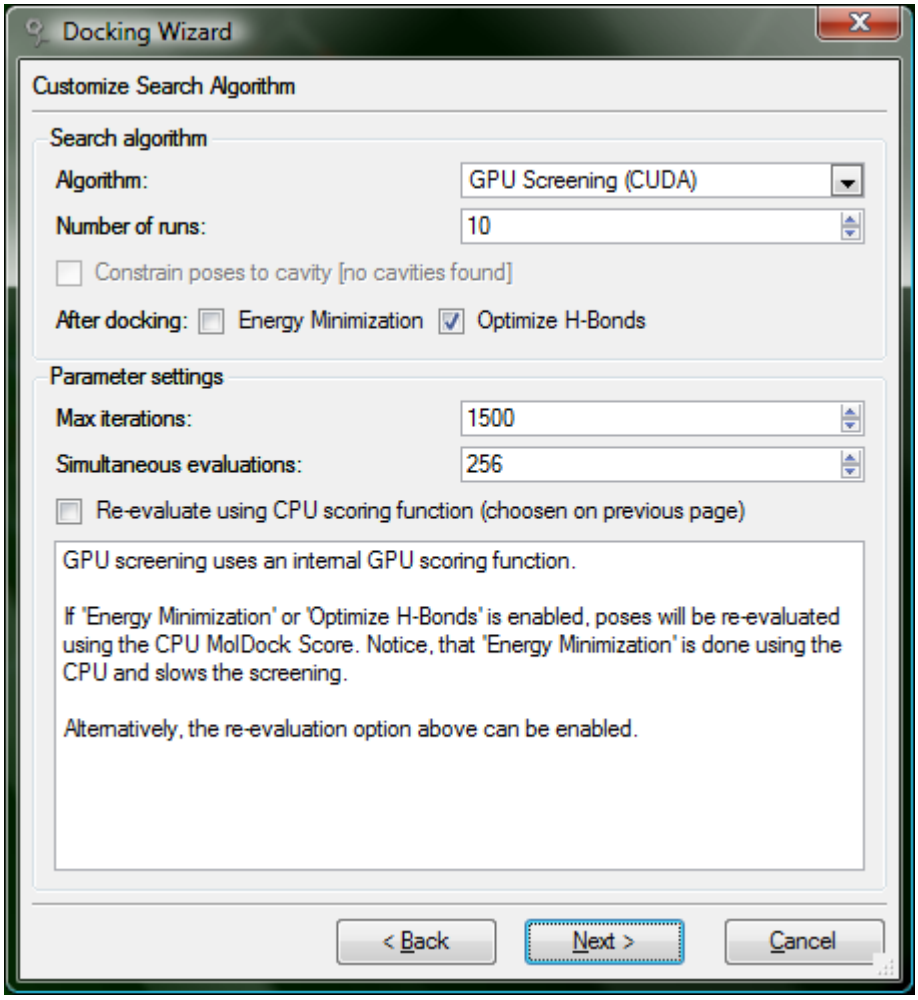

Docking Wizard
Customize Search Algorithm
Search algorithm
Algorithm:
GPU Screening (CUDA)
Number of runs:
10
Constrain poses to cavity [no cavities found]
After docking:
Energy Minimization
Optimize H-Bonds
Parameter settings
Max iterations:
1500
Simultaneous evaluations:
256
Re-evaluate using CPU scoring function (choosen on previous page)
GPU screening uses an internal GPU scoring function.
If 'Energy Minimization' or 'Optimize H-Bonds' is enabled, poses will be re-evaluated using the CPU MolDock Score. Notice, that 'Energy Minimization' is done using the CPU and slows the screening.
Alternatively, the re-evaluation option above can be enabled.
< Back
Next >
Cancel

Figure 75: Customizing settings for GPU-based screening.

It is important to notice that in contrast to the other search algorithms, the GPU screening algorithm does not use the scoring function specified on the scoring function tab. Instead it uses a PlantsPLP-like scoring function during the docking search. It is, however, possible to apply a CPU scoring function (as specified from the scoring function tab) after the docking to rerank the results. This can be enabled by checking the 'Re-evaluate using CPU scoring function (chosen on previous page)'.

The parameters for a GPU Screening are described in the next section.

The GPU Screening Algorithm

The GPU Screening algorithm uses a parallel implementation of the Nelder-Mead search algorithm to search the conformational space. Initially, a population of conformations for the current ligand is created, with the poses located on the grid points predicted by the cavity detection (at least one atom of each pose is located on a grid point). The size of the population is controlled by the **Simultaneous evaluations** parameter. Notice, that in order to properly utilise the GPU, a reasonable number of poses must be processed in parallel: usually, the default value of 256 is sufficient, but for higher-end graphics cards, it might be possible to increase the number without affecting performance. After the initial population has been constructed each pose is being minimized using the Nelder-Mead optimization technique. The **Max iterations** parameter determines how many Nelder-Mead minimization steps should be performed (if the minimization of a pose fails to improve beyond a given threshold, the pose is re-initialized with a random configuration on the cavity grid).

Using GPU Screening in a MVD Script

The GPU Screening algorithm can be specified in a mvdscript using the 'OPTIMIZERTYPE' command, e.g.:

```
....
OPTIMIZERTYPE CUDA
OPTIMIZER poses=256;steps=1500;reevaluate=false
....
```

Optionally, if more than one cuda-device is present in a machine, the desired device may be specified (before setting the optimizer type and starting the docking):

```
CUDADEVICE 1
```

Limitations of GPU Screening

Because of the different programming model for the GPU, there are certain limitations, when doing GPU Screening. The following features can not be used when docking with GPU Screening:

- Constraints are ignored during docking (both hard and soft constraints - but notice constraints are taken into account, when re-evaluating using the CPU scoring function).
- Sidechain flexibility can not be enabled.
- Displacable water is ignored.
- No support for Molegro Virtual Grid, but notice that it is possible to use GPU Screening together with KNIME.

7 Analyzing the Docking Results

7.1 Pose Organizer

The **Pose Organizer** is used to inspect poses found (see Figure 76). It allows you to browse the list of current poses, to see detailed information about specific energy contributions, to visualize hydrogen bonds, electrostatic interactions, and to calculate ranking scores and estimate binding affinity energies.

The **Pose Organizer** can be invoked in several ways. It is automatically displayed after a docking result file (with `mvdresults` file extension) has been imported to MVD by dragging-and dropping the file into MVD, using **File** | **Import Docking Results (*.mvdresults)...**, or by dragging and dropping the DockingResults icon (located in the lower left corner of the **Molegro Virtual Docker Batchjob** dialog) onto the MVD application.

Otherwise it can be invoked by using the context menu on the **Poses** category in the **Workspace Explorer** or using **Docking** | **Pose Organizer** if poses are present in the **Workspace Explorer**.

When the **Pose Organizer** is invoked it displays a list of poses parsed from the `mvdresults` file (or poses currently in the workspace). The table in the middle of the dialog window shows various columns with information about different energy contributions and other data for each pose. The columns can be changed under the **Settings** tab pane. A panel in the bottom of the dialog (**Sorting Criteria**) allows the user to sort the table by up to three different criteria.

By default the table in the middle supports multiple selection, i.e. more than one pose can be highlighted. Only highlighted poses will be visible in the 3D window. This setting is useful for quick comparison of different poses.

This default behavior can be changed by selecting **Dynamic update (notice: disables multiple poses selection)**. In this mode only one pose is shown at a time. In return it offers the possibility to visualize different interactions for the current selected pose (e.g. hydrogen bonds).

Even though **Dynamic Update** is a single-selection mode, it is possible to *lock* poses which keeps them visible even when not selected. A pose can be locked by using the context menu on its entry in the table and selecting **Lock** or **Unlock**. Locking is purely a visualization aid, and has no other consequences for the pose.

When inspecting poses obtained from different ligands, the **Only show top ...** option can be used to focus on the most promising poses for each ligand. The selection of the top poses are based on the currently chosen **Sorting criteria**.

Pressing the **Open checked poses in Molegro Data Modeller...** button makes it possible to further inspect poses using the Molegro Data Modeller (introduced in Chapter 13).

Notice: A detailed energy analysis is available by right-clicking poses in the table and selecting **Open Ligand Energy Inspector...** (see Chapter 7.3). Additional options are available in the context menu allowing the user to select, remove, and export poses. These options are also available from the **File** and **Edit** menus located in the **Pose Organizer** dialog.

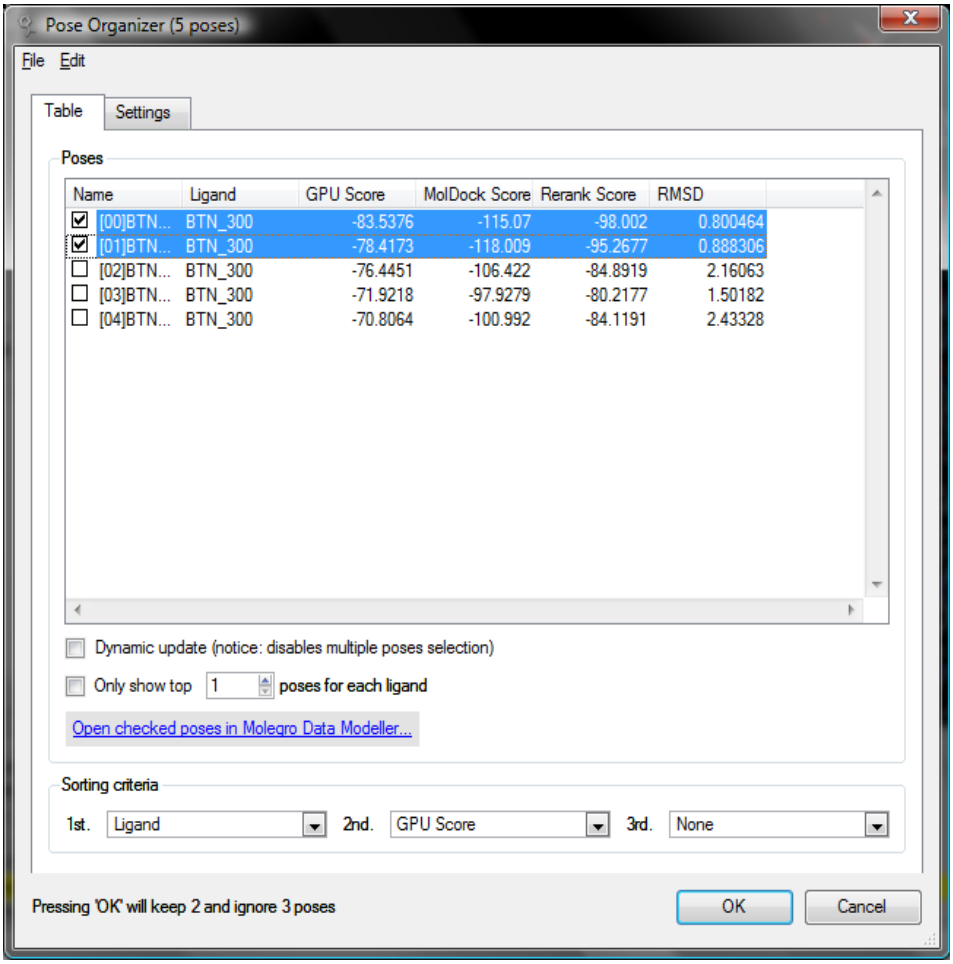

Pose Organizer (5 poses)
File Edit
Table Settings
Poses
Name Ligand GPU Score MolDock Score Rerank Score RMSD
[00]BTN... BTN_300 -83.5376 -115.07 -98.002 0.800464
[01]BTN... BTN_300 -78.4173 -118.009 -95.2677 0.888306
[02]BTN... BTN_300 -76.4451 -106.422 -84.8919 2.16063
[03]BTN... BTN_300 -71.9218 -97.9279 -80.2177 1.50182
[04]BTN... BTN_300 -70.8064 -100.992 -84.1191 2.43328
Dynamic update (notice: disables multiple poses selection)
Only show top 1 poses for each ligand
Open checked poses in Molegro Data Modeller...
Sorting criteria
1st. Ligand 2nd. GPU Score 3rd. None
Pressing 'OK' will keep 2 and ignore 3 poses
OK Cancel

Figure 76: The Pose Organizer dialog.

The **Settings** Tab Pane of the **Pose Organizer** can be used to customize the **Pose Organizer** (see Figure 77).

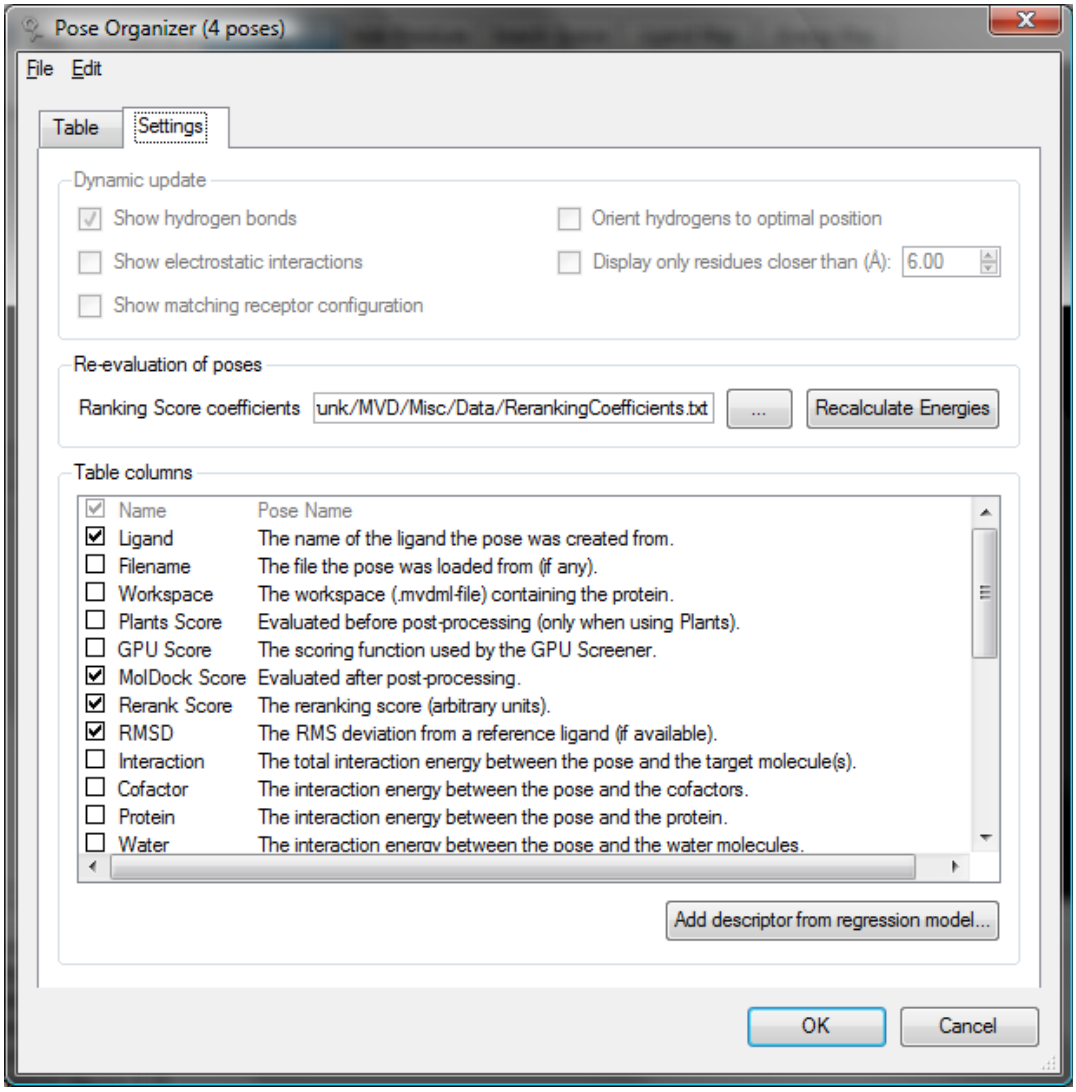

Pose Organizer (4 poses)
File Edit
Table Settings
Dynamic update
Show hydrogen bonds
Orient hydrogens to optimal position
Show electrostatic interactions
Display only residues closer than (Å): 6.00
Show matching receptor configuration
Re-evaluation of poses
Ranking Score coefficients unk/MVD/Misc/Data/RerankingCoefficients.txt ...
Recalculate Energies
Table columns
Name Pose Name
Ligand The name of the ligand the pose was created from.
Filename The file the pose was loaded from (if any).
Workspace The workspace (.mvdml-file) containing the protein.
Plants Score Evaluated before post-processing (only when using Plants).
GPU Score The scoring function used by the GPU Screener.
MolDock Score Evaluated after post-processing.
Rerank Score The reranking score (arbitrary units).
RMSD The RMS deviation from a reference ligand (if available).
Interaction The total interaction energy between the pose and the target molecule(s).
Cofactor The interaction energy between the pose and the cofactors.
Protein The interaction energy between the pose and the protein.
Water The interaction energy between the pose and the water molecules.
Add descriptor from regression model...
OK
Cancel

Figure 77: Pose Organizer settings.

The Dynamic Update Panel

The top panel (**Dynamic update**) chooses how the **Pose Organizer** behaves when single pose selection (**Dynamic update**) is enabled. It allows you to visualize hydrogen bonds, electrostatic interactions, orient hydrogens in the protein and ligand to their optimal position, and dynamically show residues close to the chosen pose. The **Orient hydrogens to optimal position** option is useful when inspecting poses as this makes it easier to see if the hydrogen bond is optimal.

Working with Receptor Conformations

When docking with sidechain flexibility a receptor conformation is saved together with each pose. When a new docking results file is imported, MVD

automatically checks whether any '.receptorConfiguration' files exist together with the poses.

If this is the case, the option **show matching receptor configuration** under **dynamic update** is enabled. When in dynamic update mode the pose organizer will now automatically change to the receptor conformation corresponding to the selected pose. If poses are imported into the workspace, their corresponding receptor conformations will automatically be added to the workspace.

The Re-Evaluation of Poses Panel

The middle panel allows for recalculation of the MolDock Score and re-ranking score terms. These scoring function values are already calculated if the poses are imported from a `mvdresults` file. Pressing the **Recalculate Energies** button will recalculate the energy terms (using the coefficients specified in the file for the re-ranking scores). Notice that the default evaluator settings will be used (e.g. internal ligand hydrogen bonds are not enabled)

The reranking score function is computationally more expensive than the scoring function used during the docking simulation but it is generally better than the docking score function at determining the best pose among several poses originating from the same ligand. The default reranking coefficients are listed in the file: `\Misc\Data\RerankingCoefficients.txt`

Binding Affinities

Predicting the experimental binding affinity of a protein - ligand complex based on a static conformation of the ligand is a difficult task. For instance, energetic contributions from solvent interactions and entropy contributions are difficult to handle in the simplified models used in molecular docking.

While the rerank-score in MVD provides an estimate of the strength of the interaction, it is not calibrated in chemical units and it does not take complex contributions (such as entropy) into account. Even though the rerank score might be successful in ranking different poses of the same ligand, it might be less successful in ranking poses of different ligands.

It is possible to create more sophisticated measures for the binding affinity using Molegro Data Modeller. New models can use the descriptors created by MVD during the docking run (the descriptors are stored in the '*.mvdresults' file). These descriptors include both terms extracted from the MolDock score function (like the protein-ligand hydrogen bonding energy) and static descriptors not using the 3D conformation of the pose (like the molecular weight or the number of nitrogen atoms).

Molegro Virtual Docker comes with a model trained to predict binding affinities. The model is located in '\Misc\Data\BindingAffinity.mdm'. The coefficients for

the binding affinity terms were derived using multiple linear regression. The model was calibrated using a data set of more than 200 structurally diverse complexes (take from the PDB data bank) with known binding affinities (expressed in kJ/mol). The Pearson correlation coefficient was 0.60 when doing 10-fold cross validation. It is important to note that this particular model was trained only on strongly interacting ligands in their optimal conformation known from the PDB complexes. Since the binding affinity measure was trained using known binding modes only, it might sometimes assign too strong binding affinities to weakly or non-binding molecules (false positives). *We therefore recommend ranking the results of a virtual screening run using the rerank score*. The binding affinity measure may then be used subsequently to get a rough estimate of the highest ranked poses.

In order to inspect the 'BindingAffinity.mdm' model, import the model into Molegro Data Modeller ('File | Import Workspace (Dataset/Models)...'). The model then appears in the workspace. By right clicking the model and selecting 'Show Details...' and choosing the 'Model' tab it is possible to see the actual multiple linear regression expression. The next section explains how to apply a model to docking results in the Pose Organizer.

The Table Columns Panel

The bottom panel (**Table columns**) determines which columns (descriptors) that are shown in the table on the first tab. Table 1 describes the descriptors that are available.

New descriptors can be added from regression models created using Molegro Data Modeller (see Chapter 13 for more details). To add a new descriptor, simply press the **Add descriptor from regression model...** button and chose the regression model from a saved *Molegro Data Modeling* (MDM) file. Notice that the regression model should only be using the same descriptors as the ones that are available in the *DockingResults* files (only valid regression models will be available in the dialog).

The Pose Organiser shows a subset of the terms in the mvdresults file as columns in the Poses table. Some of the terms use the same terminology as in the mvdresults file (specifically Name, Ligand, Filename, Workspace, RerankScore, Torsions, RMSD, MW, LE1, LE3, Hbond, Similarity Score, Electro, Hbond and Heavy Atoms), but a few terms are renamed (in order to better fit the column layout and for clarity).

Column Name	Description
Name	The internal name of the pose (a concatenation of the pose id and ligand name)
Ligand	The name of the ligand the pose was created from
Workspace	The workspace (.mvdml file) containing the protein.
Filename	The file the pose is stored as (only available when inspecting docking results from a mvdresults file)
MolDock Score	Evaluated after post-processing. [This is the 'Energy' term in a mvdresults file]
Rerank Score	The reranking score (arbitrary units)
Plants Score	Evaluated before post-processing (only when using Plants). [This is the 'PlantsScore' term in a mvdresults file]
RMSD	The RMS deviation from a reference ligand (if available)
Interaction	The total interaction energy between the pose and the target molecule(s) [This is the 'E-Inter total' term in a mvdresults file]
Cofactor	The interaction energy between the pose and the cofactors [This is the 'E-Inter (cofactor – ligand)' term in a mvdresults file]
Protein	The interaction energy between the pose and the protein [This is the 'E-Inter (protein - ligand)' term in a mvdresults file]
Water	The interaction energy between the pose and the water molecules [This is the 'E-Inter (water – ligand)' term in a mvdresults file]
Internal	The internal energy of the pose [This is the 'E-Intra (tors, ligand atoms)' term in a mvdresults file]
Torsions	The number of (chosen) rotatable bonds in the pose
Soft Constraints	The energy contributions from soft constraints [This is the 'E-Soft Constraint Penalty' term in a mvdresults file]
Electro	Short-range electrostatic protein-ligand interations (r<4.5Å)
ElectroLong	Long-range electrostatic protein-ligand interations (r>4.5Å)
HBond	Hydrogen bonding energy
Heavy Atoms	Number of heavy atoms in ligand
MW	Molecular weight (in dalton)
LE1	Ligand Efficiency 1: MolDock Score divided by Heavy Atoms count
LE3	Ligand Efficiency 3: Rerank Score divided by Heavy Atoms count

Column Name	Description
Docking Score	Evaluated before post-processing (either Plants or MolDock). [This is the 'PoseEnergy' term in a mvdresults file]
Similarity Score	The similarity score if docking with templates
DisplacedWater	The energy contributions from non-displaced and displaced water interactions (if enabled)
SMILES	Contains connectivity information - useful for 2D depictions

Table 1: Column names available in the Pose Organizer dialog.

7.2 Saving Molecules and Solutions Found

Saving Workspace

After importing and preparing molecules, all information can be saved in a MVD Workspace (MVDML) file, which contains all relevant information (position of atoms, charges, hybridization, bond orders, ligand flexibility, ...). To save a workspace, select **File** | **Save Workspace As...**. Alternatively, use the keyboard shortcut **Ctrl-S**.

Notice: Visualization objects (surfaces, labels, interactions, ...) are not saved in MVDML files.

Exporting Molecules

The **Export Molecules** dialog can be used to export all (or a selection of) the molecules available in the workspace (see Figure 78).

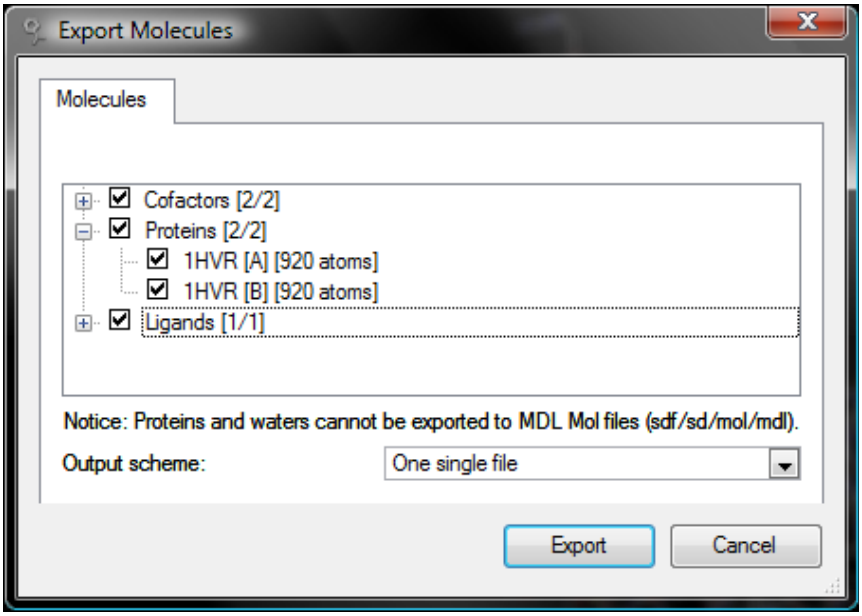


Figure 78: Export Molecules dialog: Select which molecules to export.

To export molecules, select **File** | **Export Molecules...** or **Export Molecules...** from the **Workspace** context menu in the **Workspace Explorer** (also available for proteins, ligands, cofactors, and poses).

Notice: Proteins and water molecules cannot be exported to SDF files.

Exporting Poses Found

To save the poses obtained from the docking runs, either use the **Export Molecules** dialog (described above) or save the poses from the **Pose Organizer** dialog.

7.3 Ligand Energy Inspector

The **Ligand Energy Inspector** allows you to get detailed information about the energy interactions for a given ligand or pose.

The **Ligand Energy Inspector** can be invoked in different ways. It can be started using the context menu in the **Workspace Explorer** by choosing **Open Ligand Energy Inspector** on any Ligand or Pose item. It can also be started from the **Pose Organizer** using the context menu on any pose entry or by selecting **Tools** | **Ligand Energy Inspector**.

Notice: the ligand energy inspector evaluates the energy of the ligand (or pose) when invoked. This means that the proteins, water molecules, and cofactors currently in the workspace are taken into account. If the workspace has been changed, the energy displayed here may not be the same as the one displayed in the Pose Organizer (since these were assigned during the docking evaluation).

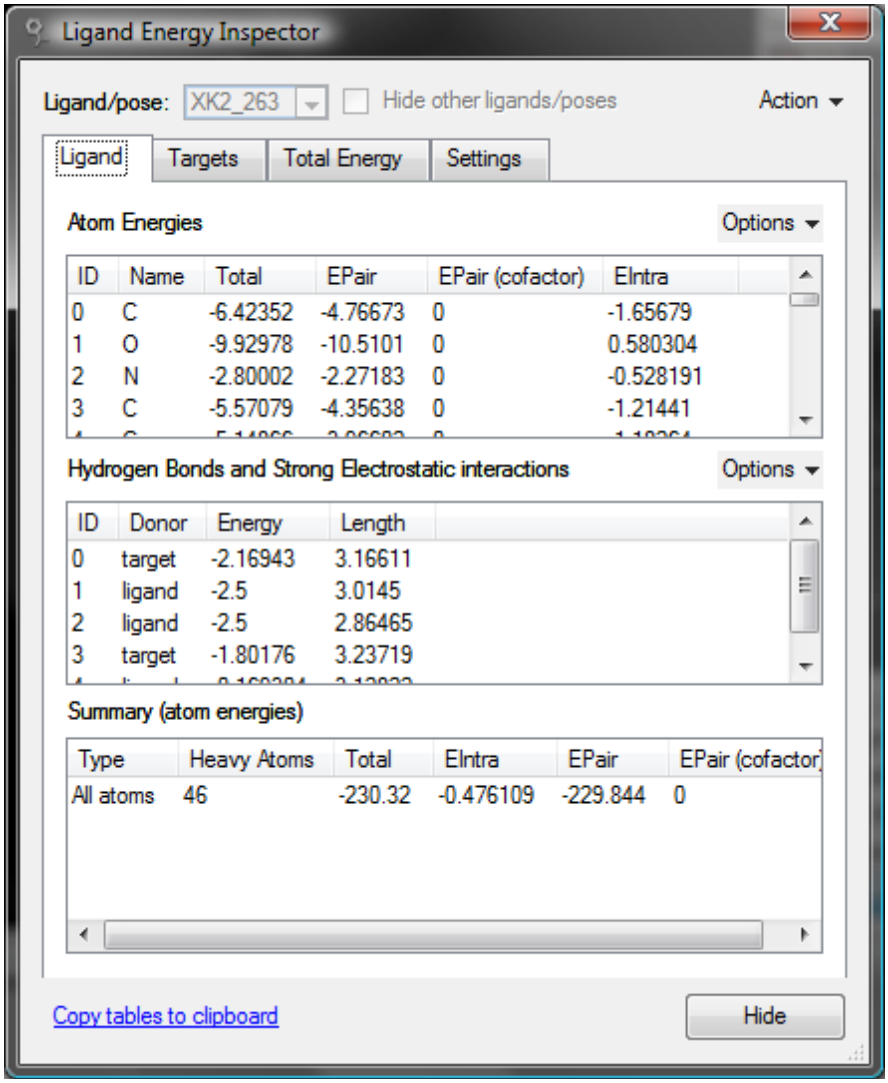


Figure 79: The Ligand Energy Inspector.

Using the Ligand/pose combo box it is possible to browse through the ligands and poses available in the Workspace. To avoid visualization of other ligands and poses when inspecting a molecule you can toggle on the Hide other ligands/poses check box.

Besides inspecting the various energy contributions, it is possible to perform various actions, using the **Action** drop down menu:

- **Style Ligand Atoms by Energy**. This will scale the radius of the ligand atoms proportionally to their energy contribution. Doing this makes it possible to get a visual overview of the important parts of the ligand.
- **Style Protein Atoms by Energy**. As above, this scales the protein atoms according to their energy contributions. Notice that protein atoms not interacting with the ligand are completely hidden. To make all protein atoms visible again, toggle the **Hide Residues** toolbar button.

- **Style Water Atoms by Energy**. This style makes it possible to get a visual overview of important interactions between water molecules and the ligand. The radius of the water atoms is scaled proportionally to their energy contributions. Water molecules with favorable interactions with the ligand are colored green and unfavorable interactions are colored red. Water molecules with no interactions to the ligand are hidden. If the **Displaceable water evaluation** option is selected, the following coloring scheme applies (see Chapter 9 for more details): displaced waters are colored yellow, non-displaced waters are colored green if they are favorable and red if they are not favorable.
- **Optimize Ligand and Protein Hydrogen Positions.** When docking with Molegro Virtual Docker the exact positions of the *rotatable* hydrogen atoms are not calculated. Instead it is assumed that the hydrogens are pointing in the optimal direction. In order to view the optimal direction of the rotatable hydrogens apply this option. Any rotatable hydrogens on the protein and ligand which are involved in hydrogen bonds will be oriented to the optimal direction.
- **Minimize Ligand.** This performs an energy minimization of the current molecule (with regard to its MolDock score energy).

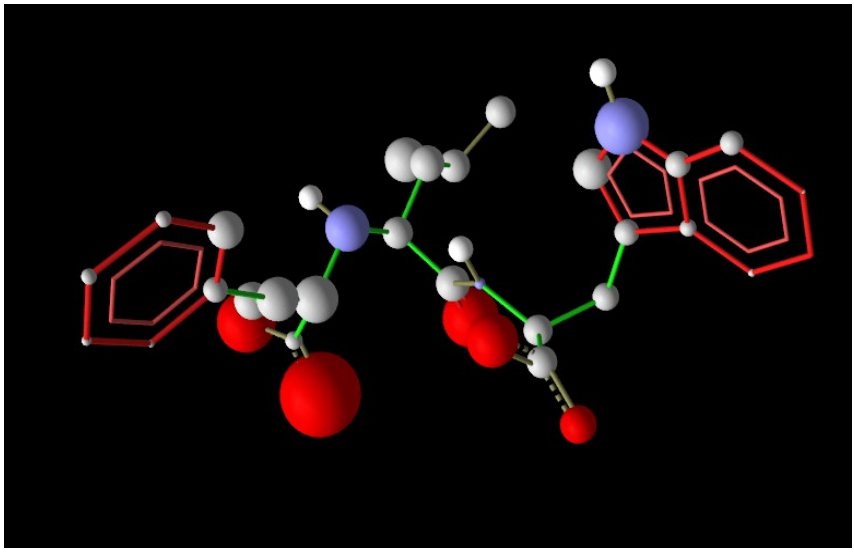


Figure 80: An example of the 'Style Ligand Atoms by Energy visualization', where atoms are scaled according to their energy contributions.

The Ligand Tab

The **Ligand** tab page consists of three tables.

The **Atom Energies** table shows information about individual atoms in the ligand. When hovering the mouse over an atom in the 3D view, it will

automatically be highlighted in the table. Similarly when selecting entries in the table, atoms will be selected in the 3D GUI. It is possible to show or hide this table using the **Options** drop-down menu.

The following types of energy contributions may be listed for a ligand atom:

- **EPair**. This is the pairwise (PLP) steric and hydrogen bonding energy between a ligand atom and a receptor atom. Pairwise interactions between a ligand and either cofactors or water molecules will show up as 'EPair (cofactor)' and 'EPair (water)'.
- **EIntra**. This is the internal ligand energy between a ligand atom and the other atoms in the ligand.
- **EElec**. This is the pairwise electrostatic interactions. For the protein they are divided into long-range and short-range interactions ('EElec (R < 4.5 Å)' and 'EElec (R > 4.5 Å)').

The second table (**Hydrogen Bonds and Strong Electrostatic Interactions**) shows a list of all hydrogen bond and strong electrostatic interactions between the ligand and the target atoms. From the **Options** drop-down menu it is possible to show or hide the table, but it is also possible to toggle the table to display covalent bonds instead (**Show Covalent Bond Energies**). Finally the **Options** menu also makes it possible to toggle whether hydrogen bonds and strong electrostatic interactions should be visualized in the GUI: Hydrogen bonds are visualized as dashed lines (where strong hydrogen bonds appear more solid) and strong electrostatic interactions are visualized as partial spheres oriented in the direction of the interaction. Green partial spheres correspond to favorable interactions, while yellow spheres correspond to non-favorable interactions.

The bottom panel (**Summary (atom energies)**) displays the sum of all atom interactions. (Notice that this is not the full energy of the ligand. Some interactions, like covalent bonding energies and constraint energies, are not included. For a complete list of energy contributions, see the **Total Energy** tab).

The Target tab

The **Target tab** displays a list of all targets atoms, residues, and molecules involved in an interaction with the inspected ligand (or pose). It is possible to switch between two views:

- **Show Residue / Molecule Contributions** which shows protein residues and water/cofactor molecules interacting with the inspected molecule.
- **Show Atom Contributions** which shows individual atoms in proteins, cofactors, and water molecules in the workspace interacting with the inspected molecule.

The atoms, residues, and molecules are only displayed in the list if the interaction energy is greater then 0.3 (in MolDock Score units).

As with the Ligand Atom Energy table, selecting atoms, residues, or molecules in the table will select them in the 3D view and vice versa. In addition, it is possible to hide non-selected residues by toggling on the **Hide Non-Selected Residues** check box.

The energy contributions are also divided into the same categories as in the Ligand Atom Table (for instance EElec and Epair).

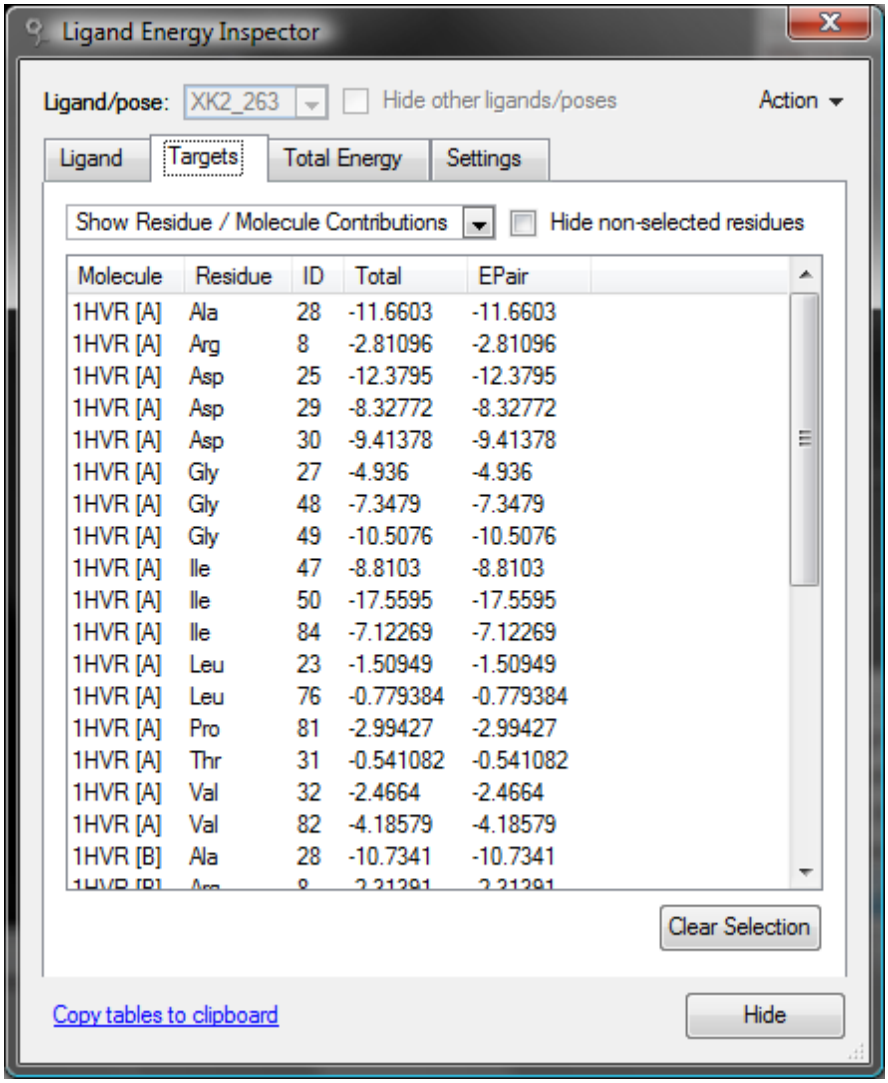

Ligand Energy Inspector
Ligand/pose: XK2_263
Hide other ligands/poses
Action
Ligand
Targets
Total Energy
Settings
Show Residue / Molecule Contributions
Hide non-selected residues
Molecule Residue ID Total EPair
1HVR [A] Ala 28 -11.6603 -11.6603
1HVR [A] Arg 8 -2.81096 -2.81096
1HVR [A] Asp 25 -12.3795 -12.3795
1HVR [A] Asp 29 -8.32772 -8.32772
1HVR [A] Asp 30 -9.41378 -9.41378
1HVR [A] Gly 27 -4.936 -4.936
1HVR [A] Gly 48 -7.3479 -7.3479
1HVR [A] Gly 49 -10.5076 -10.5076
1HVR [A] Ile 47 -8.8103 -8.8103
1HVR [A] Ile 50 -17.5595 -17.5595
1HVR [A] Ile 84 -7.12269 -7.12269
1HVR [A] Leu 23 -1.50949 -1.50949
1HVR [A] Leu 76 -0.779384 -0.779384
1HVR [A] Pro 81 -2.99427 -2.99427
1HVR [A] Thr 31 -0.541082 -0.541082
1HVR [A] Val 32 -2.4664 -2.4664
1HVR [A] Val 82 -4.18579 -4.18579
1HVR [B] Ala 28 -10.7341 -10.7341
Clear Selection
Copy tables to clipboard
Hide

Figure 81: Targets tab page.

The Total Energy Tab

The **Total Energy** tab displays a hierarchical breakdown of the various energy contributions.

When using the PLANTS scoring function, the following columns are shown:

The **Value** column displays the various terms which the PLANTS Score is based on.

The **PLANTS Score** column shows how the PLANTS score energy is composed. The PLANTS score is a sum of a subset of the Value terms (all terms are given the same weight).

For the MolDock scoring function, the following columns are available:

The **Value** column displays the various terms which the MolDock Score and the Rerank Score are based on.

The **MolDock Score** column shows how the MolDock score energy is composed. The MolDock score is a sum of a subset of the Value terms (all terms are given the same weight).

The **Rerank Score** uses a weighted combination of the terms used by the MolDock score mixed with a few addition terms (the Rerank Score includes the *Steric (by LJ12-6)* terms which are Lennard-Jones approximations to the steric energy – the MolDock score uses a piecewise linear potential to approximate the steric energy). The coefficients for the weighted Rerank Score are given in the **Rerank Weight** column, and the weighted terms and their summations are given in the **Rerank Score** column.

The relation between the terms showed in the Ligand Energy Inspector and the terms found in a mvdresults file is shown in the table below:

Ligand Energy Inspector Term	**MVDResults Term**
Total Energy	
External Ligand interaction	
Protein - Ligand interactions	
Steric (by PLP)	Steric
Steric (by LJ12-6)	VdW (LJ12-6)
Hydrogen bonds	HBond
Hydrogen bonds (no directionality)	NoHBond90
Electrostatic (short range)	Electro
Electrostatic (long range)	ElectroLong
Cofactor - Ligand	E-Inter (cofactor - ligand)
Steric (by PLP)	Not present in the mvdresults file, but can be calculated as: E-Inter (cofactor - ligand) - Cofactor (hbond) - Cofactor (elec)
Steric (by LJ12-6)	Cofactor (VdW)

Hydrogen bonds	Cofactor (hbond)
Electrostatic	Cofactor (elec)
Water - Ligand interactions	E-Inter (water - ligand)
Displacable Water interactions	E-DisplacedWater
Internal Ligand interactions	E-Intra (tors, ligand atoms)
Torsional strain	E-Intra (tors)
Torsional strain (sp2-sp2)	E-Intra (sp2-sp2)
Hydrogen bonds	E-Intra (hbond)
Steric (by PLP)	E-Intra (steric)
Steric (by LJ12-6)	E-Intra (vdw)
Electrostatic	E-Intra (elec)
Search Space Penalty	E-Penal
Soft Constraint Penalty	E-Soft Constraint Penalty

The Settings Tab

On the settings tab, the ligand evaluation can be customized. This can be important when inspecting poses from a docking run: Since the Ligand Energy Inspector is not aware of which scoring function settings were used during the docking, it is necessary to match the settings here to those selected in the Docking Wizard.

The scoring function combo box allows to choose between the docking scoring functions available in MVD: MolDock Score and PLANTS Score. For MolDock Score, the following options are available:

Internal ES toggles whether internal electrostatic interactions should be calculated for a pose, **Internal Hbond (no directionality)** toggles whether a pose should be allowed to have internal hydrogen bonds (notice that hydrogen bond directionality is not taken into account for internal hydrogen bonds in ligands), and **Sp2-Sp2 Torsions** determines whether an additional dihedral term should be added for taking Sp2-Sp2 bonds into account (see Appendix I: MolDock Scoring Function).

It is also possible to toggle on **Displaceable water evaluation** (and set the corresponding **entropy** reward) if that option was used during docking. See Chapter 9 for more details about the displaceable water model used in MVD and the additional information available in the Ligand Energy Inspector.

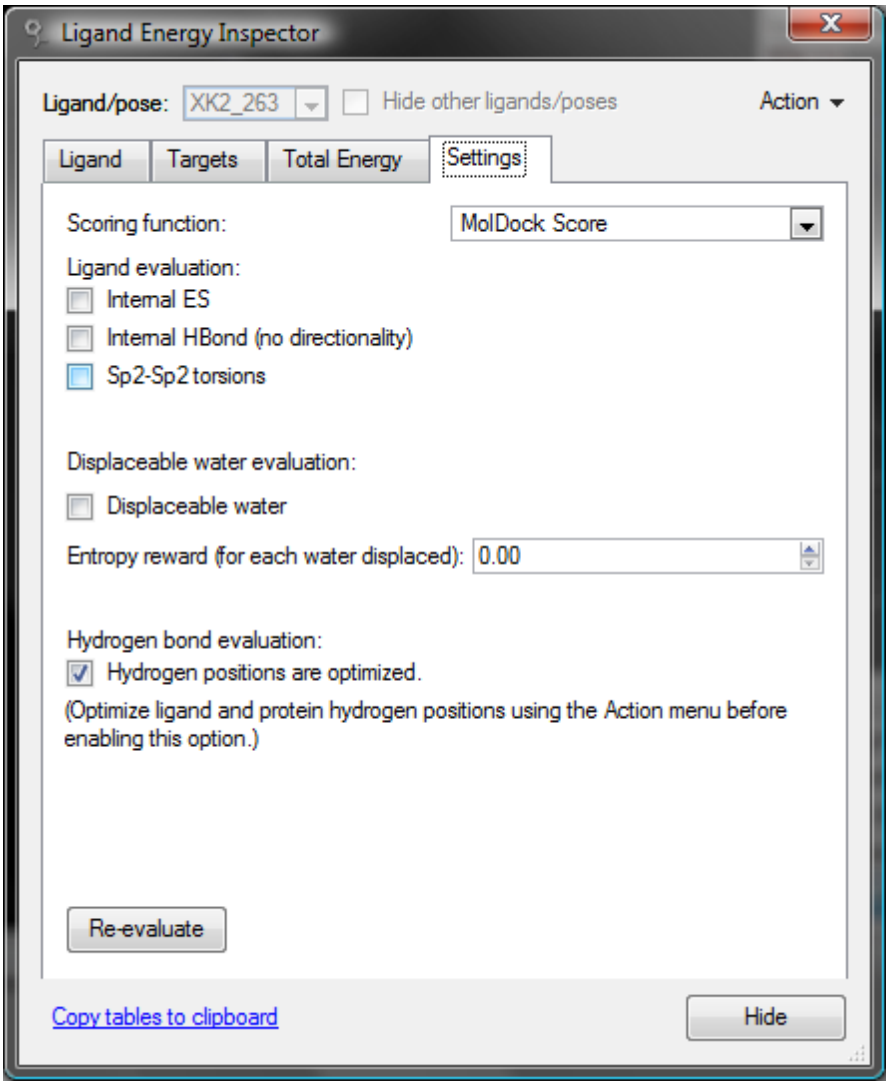


Figure 82: Settings tab page for MolDock Score.

The last option relates to hydrogen bond evaluation. When estimating hydrogen bonds, MVD does not automatically assume that rotatable hydrogen bond donors have their hydrogen atoms positioned correctly. However, if the hydrogen positions have been optimized (using **Action** | **Optimize Ligand and Protein Hydrogen Positions**) enable this option to take the full geometry of the hydrogen bond into account.

For the PLANTS Score, the following options are available:

Include hydrogens in torsion term toggles whether or not hydrogens should be included when calculating the Tripos torsion potential (see Appendix II:PLANTS Scoring Function for details about the PLANTS scoring function).

The **Use original Plants setup** option toggles between original Plants setup (using PLANTS specific binding penalty terms and ignoring entries with 'dummy' Tripos atom types in Tripos torsion potential) and MVD

implementation of PLANTS score (using another binding penalty term and including 'dummy' Tripos atom types in Tripos torsion potential). See Appendix II:PLANTS Scoring Function for details about the different binding penalty terms available for the PLANTS scoring function.

It is also possible to toggle on **Displaceable water evaluation** (and set the corresponding **entropy** reward) if that option was used during docking. See Chapter 9 for more details about the displaceable water model used in MVD and the additional information available in the Ligand Energy Inspector.

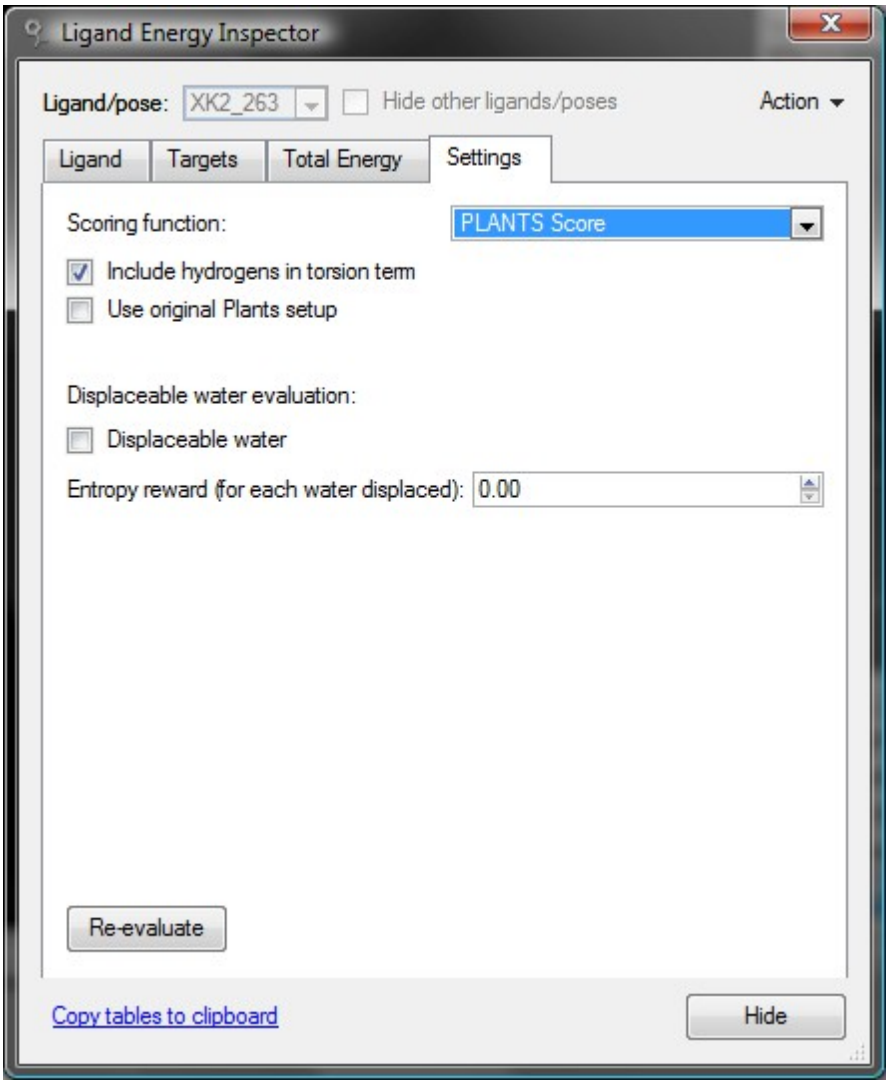


Figure 83: Settings tab page for PLANTS Score.

7.4 Ligand Map (2D Depictions)

The Ligand Map makes it possible to depict molecules (ligands and poses) in the workspace in 2D. This makes it easier to inspect the molecules, make selections, and to analyze receptor interactions. The Ligand Map can be toggled on and off using the **Ligand Map** button on the tool bar in the main

window.

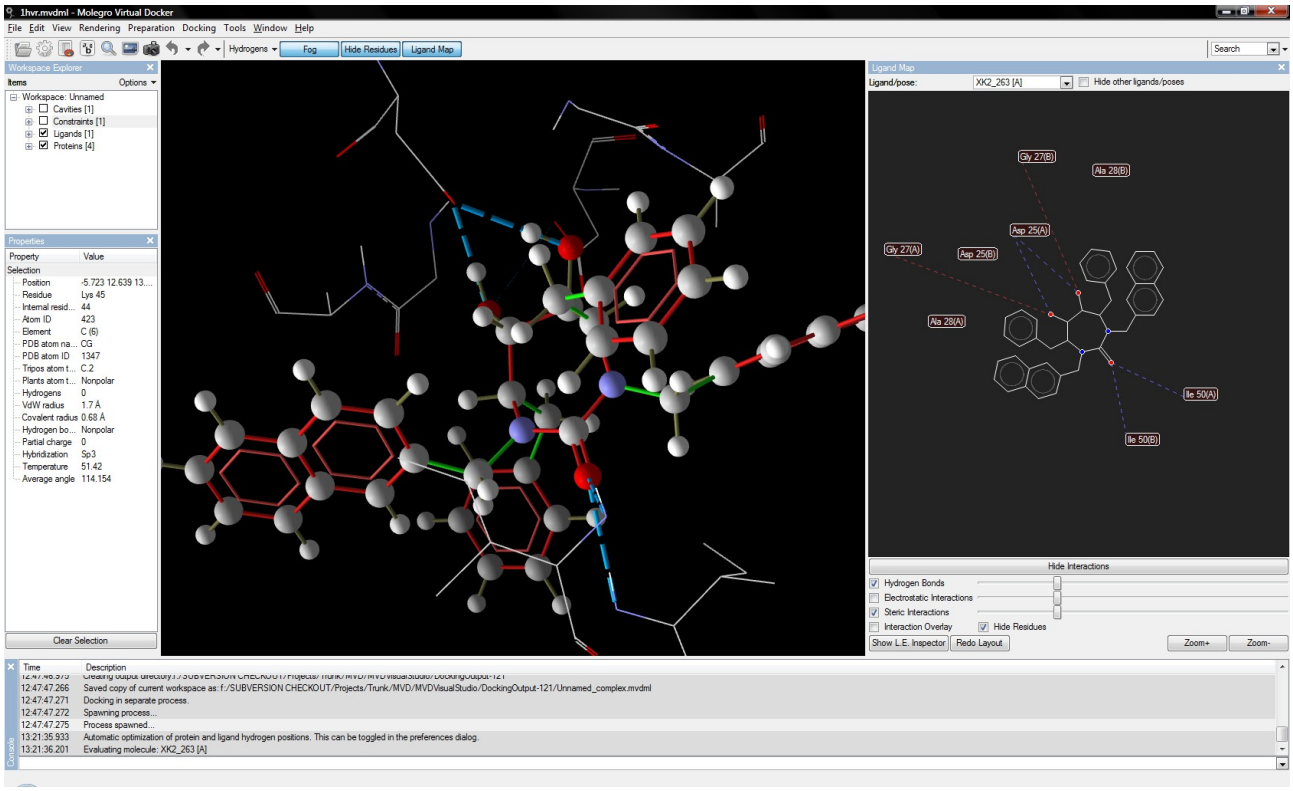


Figure 84: The Ligand Map window.

At the top of the Ligand Map window, it is possible to choose between the currently shown molecule and whether to hide other ligands and poses.

It is possible to select atoms synchronously in the 2D and 3D window by clicking on them. It is also possible to invoke the standard context menu, by right clicking on an atom. This makes it possible to e.g. change atom properties or create constraints.

Interactions and the Ligand Map

By clicking on the **Show Interactions** map, the interactions between the current ligand/pose and the receptor is shown. These interactions are the ones reported by the Ligand Energy Inspector. It is possible to press the **Show L.E. Inspector** button, which will open the **Ligand Energy Inspector**, and make it possible to adjust the scoring function settings or change the scoring function.

By default, only hydrogen bond interactions are shown. It is possible to show electrostatic interactions and steric interactions as well, by checking the respective checkboxes. It is also possible to set a minimum interaction threshold for each type of interaction. Raising the threshold slider limits the number of interactions shown. The specific value of the minimum interaction

threshold will be displayed in the statusbar of the main window, when adjusting the sliders. Notice, that for steric interactions, only non-favorable interactions (clashes) are shown - showing the numerous positive interactions would clutter the interaction diagram. However, by placing the mouse cursor over an atom or residue, the favorable steric interactions will also be shown.

It is also possible to visualize how much each ligand atom contributes to the overall binding interaction. By clicking **Interaction Overlay**, a sphere centered at each atom visualizes the strength of the interactions for this specific atom. By enabling the **Hide Residues** option it is possible to hide residues in the 3D visualization window that are not shown in the 2D Ligand Map.

The **Redo Layout** button makes it possible to calculate a new layout for the molecule and its interactions, for instance if the layout contains clashing bonds.

It is possible to zoom in and out using either the mouse wheel, or the zoom buttons in the lower right corner of the window.

7.5 Pose Modifier

It is possible manually to modify a ligand (or a pose found) by right-clicking the molecule in the **Workspace Explorer** and selecting **Modify Pose** (see Figure 85). When invoking the **Pose Modifier**, a new pose is created.

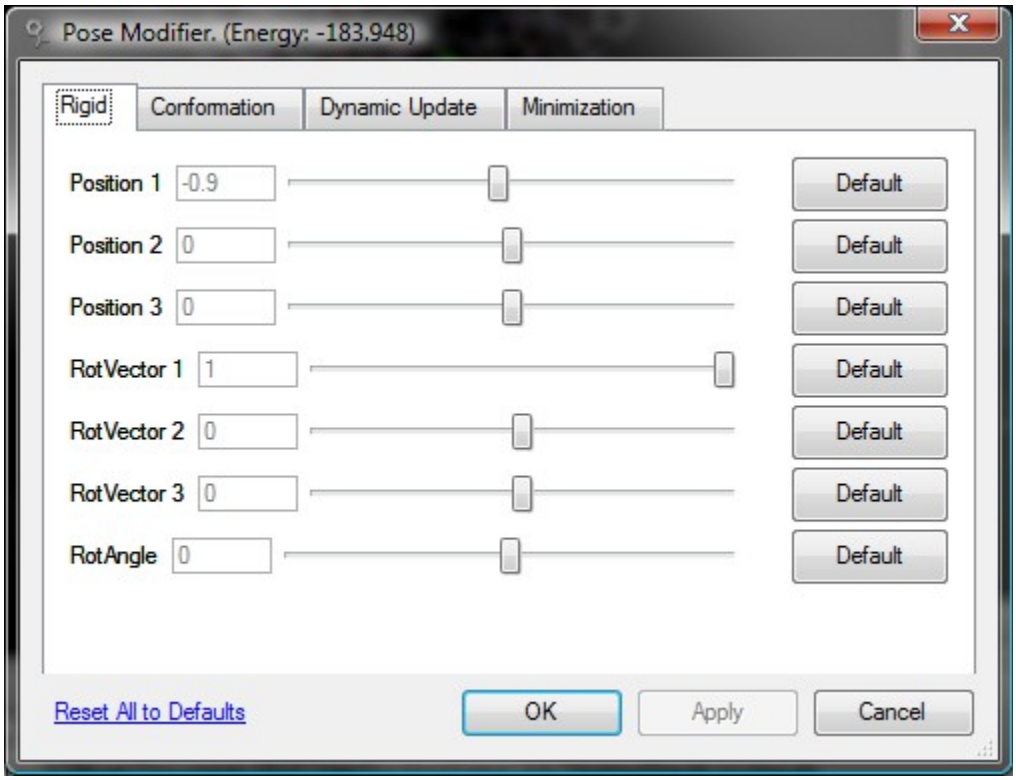


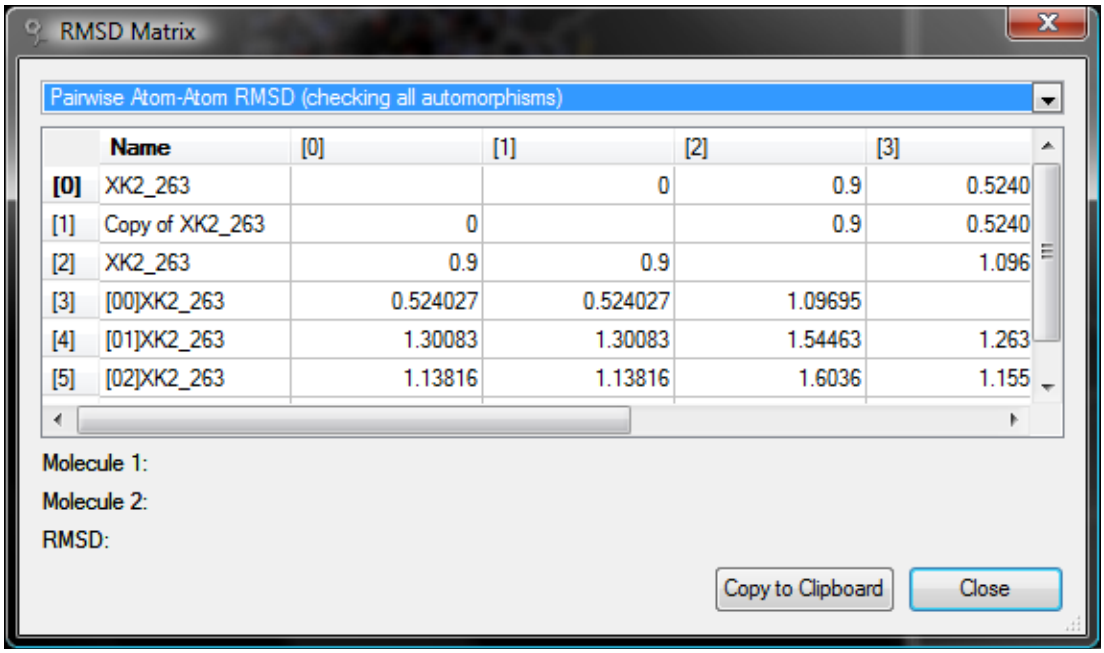
Figure 85: Pose Modifier dialog.

Notice: It is not possible to directly modify poses after the workspace has been

saved and reloaded. However, ligands can be modified any time. To modify poses saved, these can be converted to ligands and modified afterwards (which will result in a new modified pose). Different interactions can also be visualized on-the-fly (**Dynamic Update** tab).

7.6 RMSD Matrix

The **RMSD Matrix** dialog can be used to quickly inspect deviations between molecules in the workspace. In addition to the standard measure **Pairwise Atom-Atom RMSD (by ID)**, two variants **Pairwise Atom-Atom RMSD (checking all automorphisms)** and **Pairwise Atom-Atom RMSD (by nearest unmatched neighbour)** of the RMSD measure tries to take intrinsic symmetries of the molecule into account when calculating RMSD. The recommended choice is **Pairwise Atom-Atom RMSD (checking all automorphisms)**, which is also used by default.



	Name	[0]	[1]	[2]	[3]
[0]	XK2_263		0	0.9	0.5240
[1]	Copy of XK2_263	0		0.9	0.5240
[2]	XK2_263	0.9	0.9		1.096
[3]	[00]XK2_263	0.524027	0.524027	1.09695	
[4]	[01]XK2_263	1.30083	1.30083	1.54463	1.263
[5]	[02]XK2_263	1.13816	1.13816	1.6036	1.155

Figure 86: RMSD Matrix dialog.

The dialog can be invoked by choosing **RMSD Matrix** from the **Tools** menu. The **Copy to Clipboard button** can be used to copy the table to the clipboard for further inspection in an external text editor or spreadsheet.

8 Sidechain Flexibility

It is possible to work with sidechain conformational changes in two ways:

- By softening the potentials (the steric, hydrogen bonding, and electrostatic forces) used during the docking simulation. This is done in order to simulate flexibility in the binding pocket ('induced fit').
- By defining which residues should be considered flexible during the docking simulation. The backbone is kept rigid, but the torsional angles in the sidechains are allowed to change.

When sidechain flexibility has been setup, the following steps are applied during the docking simulation:

- The ligands will be docked with the softened potentials. At this point the receptor is kept rigid at its default conformation.
- After each ligand has been docked, the sidechains chosen for minimization will be minimized with respect to the found pose. After repositioning the sidechains, the ligand will be energy minimized. The repositioning of the sidechains and minimization of the ligand will be performed using the standard non-softened potentials.

It is preferable to use the 'Tabu Clustering' algorithm in order to ensure a greater diversity of the found poses during the docking simulation (see Section 6.3). Also notice that only the 'MolDock [GRID]' potential supports softened potentials. The 'MolDock' scoring function will always use unsoftened potentials. The **Docking Wizard** will warn you if either of these requirements are not fulfilled.

When sidechain flexibility has been setup a *sidechain flexibility description* is added to the workspace. This information is stored as part of the MVDML file.

In the **Workspace Explorer** a new category (**Flexible Residues**) will appear, indicating that a sidechain flexibility description is present in the workspace.

8.1 The Setup Sidechain Flexibility Dialog

To invoke the **Setup Sidechain Flexibility** dialog, select **Docking** | **Setup Sidechain Flexibility** (see Figure 87). If the workspace already contains a Sidechain flexibility description, you can edit it by using the context menu on the **Flexible Residues** group in the **Workspace Explorer** and selecting **Setup Sidechain Flexibility**.

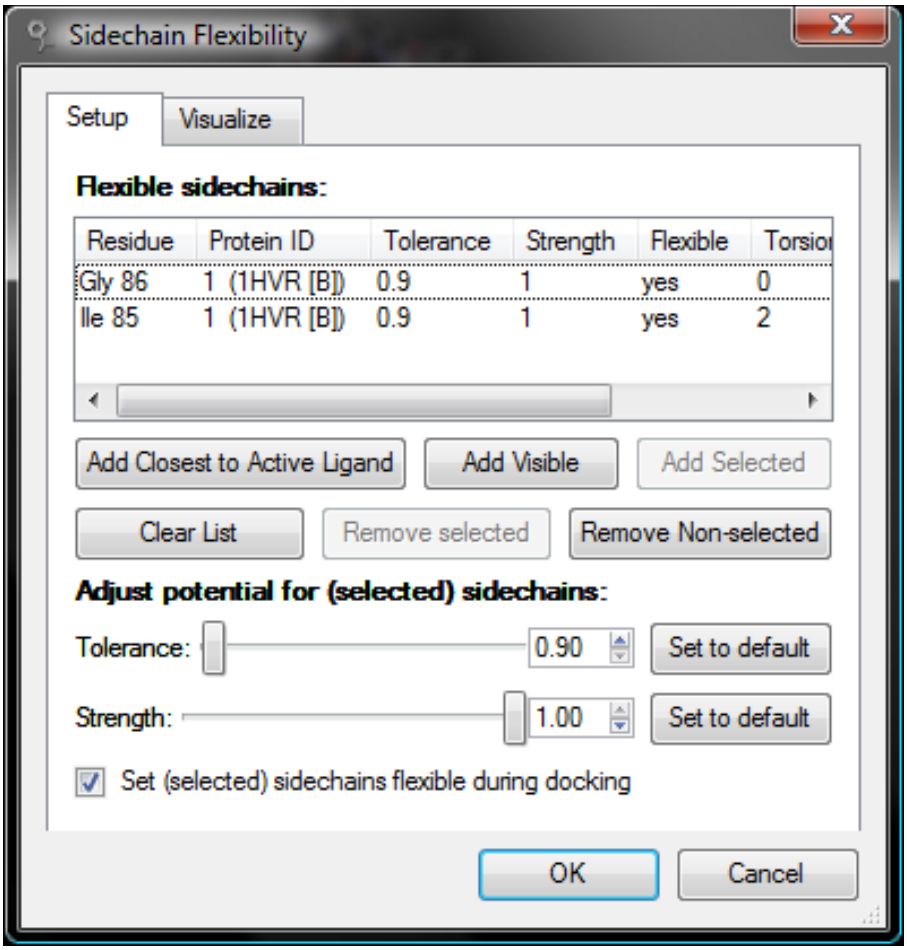


Figure 87: The Sidechain Flexibility dialog.

The **Setup** tab allows you to select a number of sidechains and define their individual properties: that is, how the potential should be softened and whether the sidechain should be allowed to be flexible during the docking or not.

Several options exist for choosing the relevant residues:

Add Closest to Active Ligand - This will choose all sidechains which are close enough to the active ligand to interact with it. More precisely: for each given sidechain, a sphere bounding all possible configurations of the sidechain is

calculated, and it is tested whether any atom in the active ligand is close enough to make a steric contact with an atom in this bounding sphere (for the 'MolDock' potential, all steric contacts are cut off at a distance of 6.0 Å). Notice that the 'Active Ligand' can be set in the Workspace Explorer window: it is the ligand which name is prepended with an '[Active]' label.

Add Visible - This will add all sidechains which are currently visible in the 3D Visualization window. This feature can be used together with the **Hide Residues** dialog where it is possible to hide sidechains depending on the distance from some given object.

Add Selected - This feature makes it possible to select sidechains directly in the 3D Visualization window. A sidechain is considered to be selected if one or more atoms inside it are chosen.

Clear List - Removes all sidechains from the list.

Remove Selected - Removes all sidechains that are currently highlighted in the sidechain list.

Remove Non-selected - Removes all sidechains that are not highlighted in the sidechain list.

Sidechains added to the list will be visualized with a wireframe sphere in the 3D Visualization window. If one or more sidechains are highlighted in the list, only this subset will be visualized.

The list of chosen sidechains contains the following information:

Residue - The residue name/id.

Protein ID - The protein or (protein chain) ID and name.

Tolerance - See below

Strength - See below

Flexible - Indicates whether the sidechain is currently selected for minimization in the docking simulation or not. By default all sidechains added to the list will be set as flexible - however it is possible to add sidechains to the list and only have their potential softened, while keeping them rigid.

Torsions - The number of degrees of freedom in the given sidechain. The degrees of freedom that are minimized during the docking simulation are the torsional angles in the sidechain.

Mean T - The temperature factor or B-factor is a measure of how much a given atom vibrates around its position in the crystallographic model. This can be useful since a high B-factor may indicate that the residue is flexible. **Mean T** is the average temperature for the (heavy) atoms in the sidechain.

Max T - The same as above, except that **Max T** is the single highest temperature factor of all (heavy) atoms in the sidechain.

The columns in the list can be toggled on and off using the context menu on the list view.

Modifying the Potential

The **Tolerance** of a potential refers to the size of the region between a ligand atom and a receptor atom where the interaction energy is optimal. For non-polar steric interactions (such as two carbon atoms) the interaction is optimal between 3.6 Å and 4.5 Å (see Appendix I: MolDock Scoring Function for more information about the scoring function). This gives a tolerance of 0.9 Å. If the tolerance is increased to e.g. 1.5 Å, the interaction would be optimal at distances between 3.3 Å and 4.8 Å. Notice that the tolerance is only softened for atoms in the sidechain, not for the backbone atoms. Also changing the tolerance only affects pairwise steric and hydrogen bonding potentials - electrostatic forces are not changed.

The **Strength** factor is multiplied onto all interaction energies for the sidechain (atomic pairwise steric interactions, hydrogen bondings and electrostatic interactions). If a sidechain is known to be very flexible, set its strength to zero in order to turn all its interactions off during the docking simulation. Notice that the strength factor does not change the interactions of the backbone atom.

After choosing a number of sidechains and configuring their flexibility options, press **OK** to add a sidechain flexibility description to the workspace. A new category will appear in the **Workspace Explorer** (**Flexible Residues**) and the selected sidechains will be indicated visually in the workspace as wireframe spheres. The sphere color will depend on the *strength* parameter, and the sphere size will reflect the *tolerance* parameter (see Figure 88).

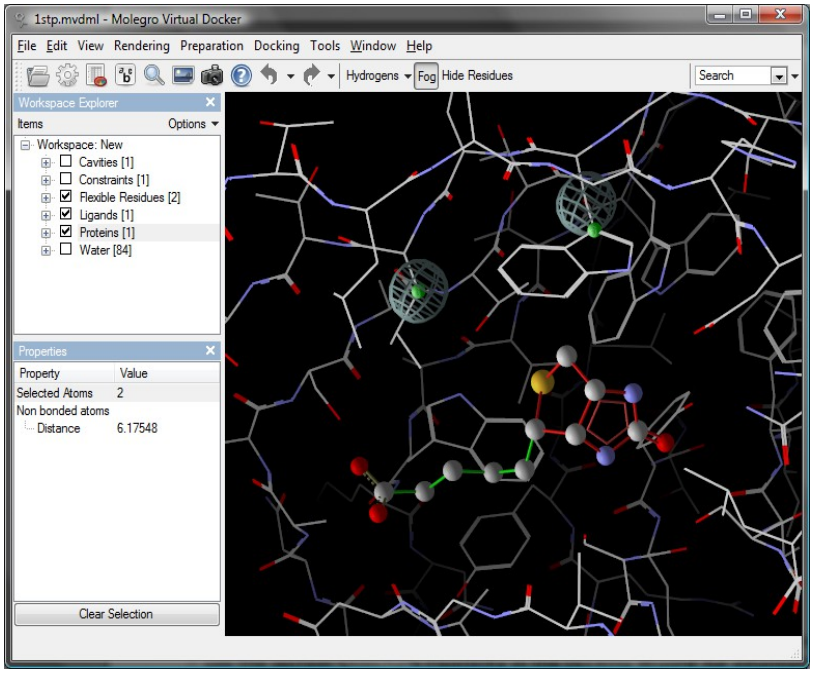


Figure 88: Visualization of flexibility descriptors.

Sidechain flexibility descriptors are saved as part of the workspace. The sidechain flexibility description is read and used directly by the docking engine – see the section 'Sidechain Flexibility in the Docking Wizard' for information about setting up a docking run with sidechain flexibility.

The Visualization Tab

The **Visualization** tab (Figure 89) is used to create small animations, showing wireframe surfaces interpolating between two different potential energy landscapes. The wireframe surfaces are determined by probing a receptor energy grid with a carbon atom. The grid surface marks the boundary between energetically favorable and non-favorable regions. (Notice that polar atoms capable of making hydrogen bonds would be allowed to be closer to the protein).

The tab serves two purposes:

- Visualizing the effects of softening the potentials, or
- Comparing the potentials of two different receptors. This can be useful if you have several different crystallographic structures and want to compare them in order to determine how the receptor potential should be softened.

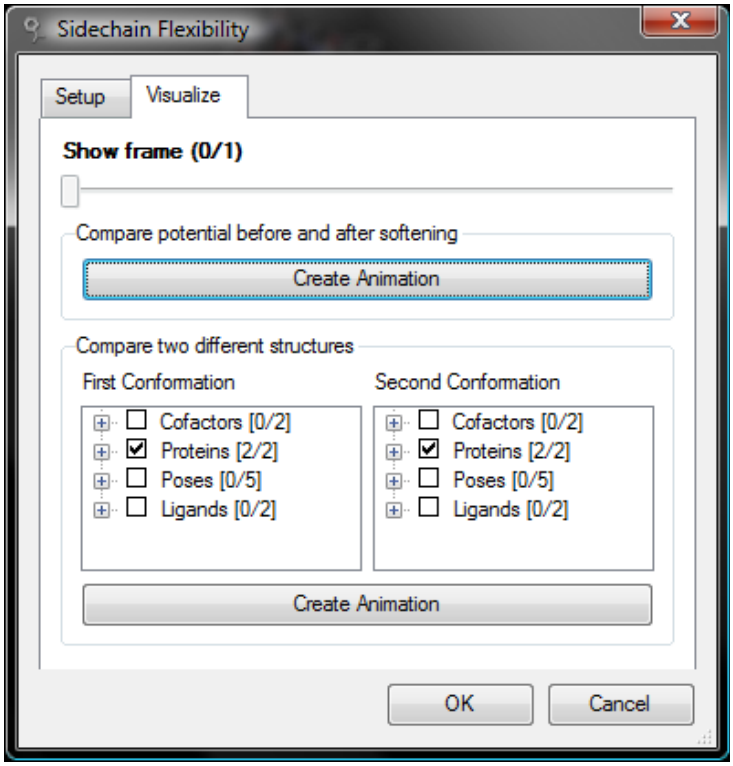


Figure 89: The Sidechain Flexibility visualization tab.

In order to visualize the effects of softening the potentials, first setup a *search space* – the surface grids will only be drawn for molecules inside the search space. If a search space has not already been defined, you can use the **Search Space** button on the main toolbar to define one.

When pressing **Create Animation**, 20 wireframe energy contour surfaces will be created. In order to inspect the surfaces (view the animation) use the slider located at the top of the dialog (**Show frame (x/x)**). After having inspected the energy changes, the surfaces can be removed from the workspace by using the context menu on the **Surfaces** category in the Workspace Explorer (**Remove All Surfaces From Workspace**).

8.2 Sidechain Flexibility in the Docking Wizard

If the **Docking Wizard** is invoked and the workspace contains a sidechain flexibility description, a new page will appear in the wizard after the first page (Figure 90: Sidechain Flexibility in the Docking Wizard.):

The first option: **Soften potentials during the docking** turns on the softening procedure for the potentials. Notice that is necessary to use the 'MolDock Score [Grid]' for potential softening to have any effect.

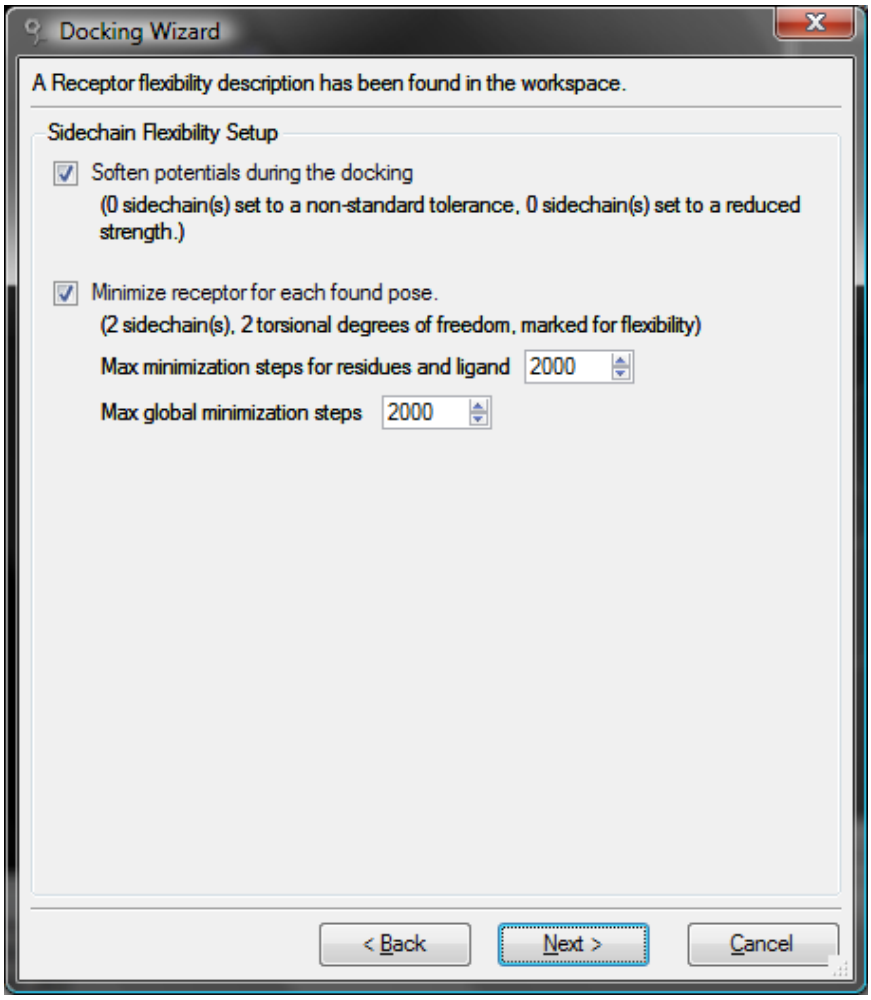

Docking Wizard
A Receptor flexibility description has been found in the workspace.
Sidechain Flexibility Setup
Soften potentials during the docking
(0 sidechain(s) set to a non-standard tolerance, 0 sidechain(s) set to a reduced strength.)
Minimize receptor for each found pose.
(2 sidechain(s), 2 torsional degrees of freedom, marked for flexibility)
Max minimization steps for residues and ligand 2000
Max global minimization steps 2000
< Back
Next >
Cancel

Figure 90: Sidechain Flexibility in the Docking Wizard.

The next option: **Minimize receptor for each found pose** turns the post-docking minimization step on for the best found solutions during the docking run. First, the flexible sidechains are reoriented taking the pose into account. Afterwards, the pose is energy minimized. It is possible to define the maximum number of global and local minimization steps. The receptor and ligand minimization is performed using the Nelder-Mead simplex algorithm, and is described in more detail in Section 3.20.

Notice that is advisable to use Tabu Clustering during the docking simulation in order to ensure a greater diversity of the returned poses. See the **Docking Wizard** (Section 6.3) for more information.

8.3 Sidechain Flexibility and Scripting

If using sidechain flexibility together with scripting, first add a sidechain flexibility description to the workspace. The actual softening of the potentials and post-docking minimization steps, can be scripted using the

'**MinimizeReceptor**=[LocalSteps,GlobalSteps]' option for the DockSettings command, and the '**SoftenPotential**=[true|false]' option for the evaluator. Notice that it is advisable to use Tabu Clustering to ensure greater diversity of the returned poses before the minimization run is executed.

This is how a typical script using sidechain flexibility might look like:

```
DOCKSETTINGS maxIterations=2000;runs=20;ignoreSimilarPoses=false;
IgnoreSimilarPosesThreshold=1;MaxPoses=5;MinimizeReceptor=2000,2000

EVALUATORTYPE MolDockGrid

EVALUATOR cropdistance=0;gridresolution=0.30;hbond90=true;
SoftenPotential=true;tabuclustering=true,2,100,id

OPTIMIZER cavity=true;popsize=50;scalingfactor=0.50;crossoverrate=0.90;
offspringstrategy=1;terminationscheme=0;earlytermination=0.01;
clusterthreshold=0.0

LOAD SomeComplex.mvdml

DOCK
```

Inspecting Docking Results

When inspecting the docking results in the **Pose Organizer**, it is possible to automatically view the receptor conformation corresponding to the selected pose. This is done by enabling **Show matching receptor configuration** under **Settings** | **Dynamic update** in the **Pose Organizer** dialog. Notice: this requires that the **Pose Organizer** is in **Dynamic Update** mode. For more information see Section 7.1.

9 Displaceable Water

Under normal circumstances good docking results may be obtained without taking explicit water molecules into account. However, sometimes water molecules can play a key role in a protein-ligand interactions by forming or mediating hydrogen bonds between the protein and the ligand. In such cases taking explicit water into account during docking may be necessary to improve the docking accuracy. However, even for a protein structure with explicit water molecules, the ligand may displace them. One way of handling this would be to manually create multiple receptor configurations with individual water molecules toggled on or off. In MVD, the **displaceable water model** makes it possible for a ligand to keep favorable and displace non-favorable water molecules during docking. The identification of key water molecules is done during the evaluation of the protein-ligand binding and is thus separated from the conformational sampling.

While the displaceable water model may be successful in some cases where ordinary docking fails it also has some restrictions (see below). Therefore, we recommend to use the displaceable water model in situations where docking without water is not successful. In addition, the model requires *a priori* knowledge of likely water molecule positions something which is not always available.

Restrictions:

- MVD cannot predict water positions in the binding site. If possible, the water molecules should be obtained from an *apo* structure since a *holo* structure containing a co-crystallized ligand might already have displaced the water molecules. Another possibility is to use other third-party software products to predict or identify relevant positions of water molecules.
- Search space: Even though no additional degrees of freedom are

introduced when using the displaceable water model in MVD, the search space may be less predictable resulting in poorer performance. If needed, increasing the number of docking runs can improve the performance.

- Speed: Enabling the displaceable water model increases the docking runtime (dependent on the number of water molecules in the workspace). Therefore, we recommend to focus on a selected subset of water molecules and remove all irrelevant water molecules from the workspace before starting the docking run (waters can be easily removed using the crop option in the **Hides Residues** dialog). Example: re-docking the ligand from 1STP (available in examples folder) including six relevant water molecules is approximately twice as slow when enabling displaceable water evaluation compared with default settings. Taking all water molecules into account makes the displaceable water evaluation eight times slower than docking with default settings.

The overall strategy when evaluating a given ligand conformation is to inspect each water molecule individually and decide whether or not it interacts favorably with the ligand. Favorable water molecules are kept whereas non-favorable water molecules are displaced or ignored. The next section describes this evaluation procedure in more details.

Displaceable Water Evaluation

The displaceable water evaluation in MVD consists of two main steps:

The first step is to pre-calculate energy interactions between a water molecule and all protein and cofactor heavy atoms: $E_{water\text{-}protein/cofactor}$ and between a water molecule and all other water molecules: $E_{water\text{-}other\ waters}$. Both $E_{water\text{-}protein/cofactor}$ and $E_{water\text{-}other\ waters}$ contributions are calculated using the MolDock scoring function (see Appendix I: MolDock Scoring Function for details).

Notice: the $E_{water\text{-}other\ waters}$ interactions are pre-calculated and include all water molecules in the workspace. In some cases, the pre-calculated $E_{water\text{-}other\ waters}$ contributions might differ a bit compared with the actual contributions from the neighbouring water molecules, since displaced water molecules are included.

The energy required to remove a water molecule is: $E_{remove\ water} = -(E_{water\text{-}protein/cofactor} + E_{water\text{-}other\ waters} + E_{entropy\ reward})$.

When a water molecule is displaced it gains rotational and translational degrees of freedom (compared with its bound state when binding to a protein or a ligand). Thus, the $E_{entropy\ reward}$ is a reward representing the gain in entropy that occurs when a water molecule is displaced since a system will always favor states with higher entropy (according to Gibbs free energy).

It can be difficult to determine the optimal entropy reward but it should be less than the contributions from a water molecule interacting with other

protein/cofactor/water atoms, i.e. $E_{entropy\ reward} < -(E_{water\text{-}protein/cofactor} + E_{water\text{-}other\ waters})$. The higher the entropy reward is the easier it gets to displace water molecules. By default, $E_{entropy\ reward} = 0$ but the entropy reward can be customized by the user. Notice that the units for the entropy reward are arbitrary (the entropy reward is not based on physical units).

The next step is to look at each water molecule during an evaluation of a ligand/pose and decide if the water molecule should be displaced or not. First, the interaction energy between all ligand atoms and the water molecule is calculated using the PLP potential: $E_{water\text{-}ligand}$. Afterwards, the water molecule is categorized into one of the following categories:

- **Ignored**: a water molecule with no net ligand interaction ($E_{water\text{-}ligand} = 0$) is simply ignored (water molecules located more than 6 angstrom from the ligand will not interact with the ligand).
- **Displaced**: a water molecule is displaced if $E_{water\text{-}ligand} > E_{remove\ water}$.
- **Non-displaced**: a water molecule is not displaced if $E_{water\text{-}ligand} \leq E_{remove\ water}$ (e.g. water molecules with favorable ligand interactions, $E_{water\text{-}ligand} < 0$, are always kept).

Using the displaceable water model, the total energy contribution can be summarized in the following formula:

$$E_{Total\,water\,energy} = \sum_{Non-displaced\,waters} E_{water-ligand} + \sum_{Displaced\,waters} -\left(E_{water-protein/cofactor} + E_{water-other\,waters} + E_{entropy\,reward}\right)$$

9.1 Docking with Displaceable Water Molecules

Docking with displaceable water molecules can be enabled from the Docking Wizard or from MVD scripts.

Docking Wizard Settings

If the **Docking Wizard** is invoked and the workspace contains one or more water molecules, the option to setup displaceable water evaluation becomes available in the scoring function tab page. To include displaceable water evaluation, toggle on the **Dispaceable Water** option. It is also possible to specify an **entropy reward** for displacing water molecules, which adds a constant reward for each water displaced.

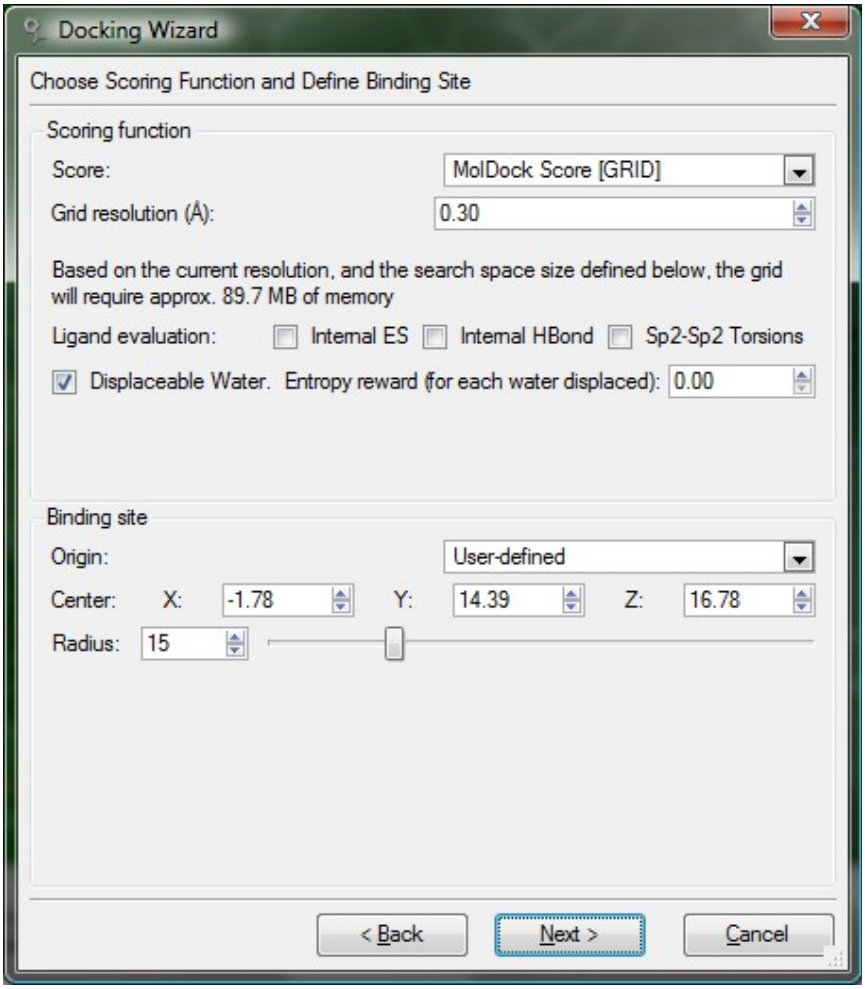


Figure 91: Docking Wizard: Enabling docking with Displaceable Water.

Since handling of displaced/non-displaced waters is done during the evaluation step only and therefore separated from the conformational search, no other settings are needed to enable displaceable waters.

Scripting Settings

It is also possible to enable displaceable water evaluation when performing batch job runs using the MVD scripting language. The **DisplaceWater**=[true|false] option is used to toggle the displaceable water evaluation on or off and the entropy reward is specified using the **DisplaceWaterReward**=[0.0-10.0] option. Both settings are specified as parameters for the **EVALUATOR** command.

This is how a typical script using using displaceable water evaluation might look like:

```
DOCKSETTINGS maxIterations=1500;runs=10;ignoreSimilarPoses=true;MaxPoses=5;
IgnoreSimilarPosesThreshold=1
```

```
EVALUATORTYPE MolDockGrid
EVALUATOR cropdistance=0;gridresolution=0.30;ligandes=false;sp2sp2bond=false;
internalhbond=false;hbond90=true;DisplaceWater=true;DisplaceWaterReward=0
OPTIMIZERTYPE MSE
OPTIMIZER populationsize=50;cavity=true;creationEnergyThreshold=100;
poseGenerator=10,10,30;recombine=true;maxsimplex=750;simplexsteps=300;simplexd
istancefactor=1;clusterthreshold=1.00;keepmaxposes=5
LOAD "SomeComplex.mvdml"
DOCK
```

9.2 Inspecting Results

After docking using the displaceable water evaluation, it is possible to inspect the docking results in the **Pose Organizer** or in the **Ligand Energy Inspector**.

Pose Organizer

When inspecting the docking results in the **Pose Organizer**, it is possible to see the overall energy contributions summarizing interactions between non-displaced waters and the ligand combined with energy contributions and entropy rewards for displaced water molecules. These contributions are listed in the **DisplacedWater** column, which can be enabled from the list of optional columns.

Ligand Energy Inspector

Using the **Ligand Energy Inspector** introduced in Section 7.3, it is possible to inspect the displaceable water evaluation in more details. In short, the Ligand Energy Inspector dialog allows for easy inspection of displaced/non-displaced waters, energy contributions from displaced/non-displaced water interactions, and styling of water molecules for visual inspection in the 3D visualization window.

Since the Ligand Energy Inspector is not aware of which scoring function settings were used during the docking run, it is necessary to match the settings selected in the Docking Wizard or specified in the MVD script file. Therefore, the **Displaceable water** option needs to be toggled on in the Settings tab (see Figure 92). Also, if any entropy reward was applied during docking the same reward value should be specified in the **Entropy reward for each water displaced** setting. Afterwards, to update the energy contributions listed in the other tab pages, the ligand needs to be re-evaluated by pressing the **Re-evaluate** button.

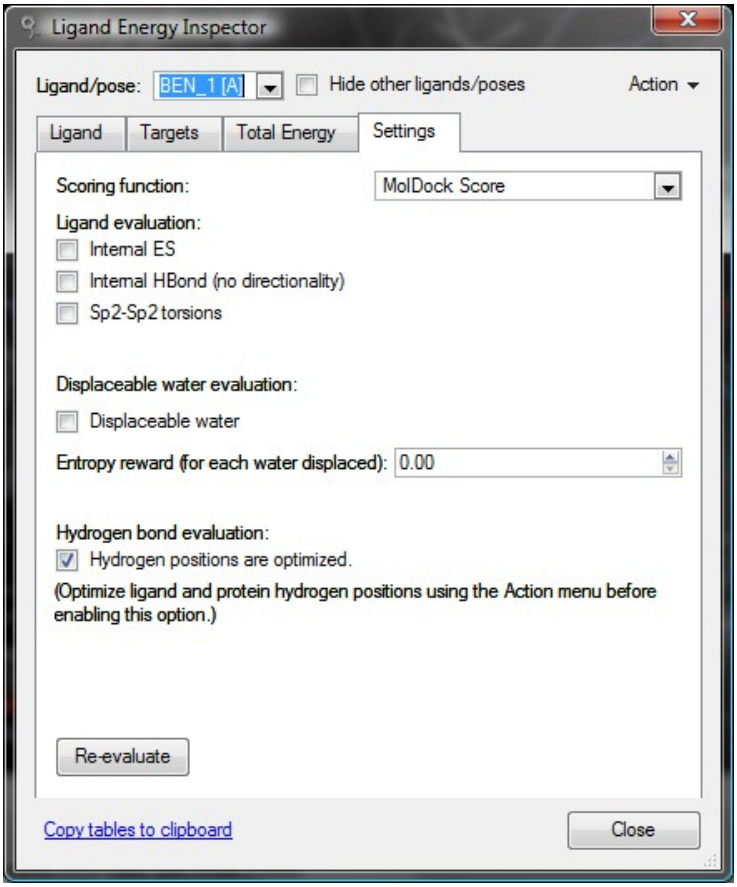


Figure 92: Ligand Energy Inspector: Re-evaluating displaceable water interactions.

When the ligand has been re-evaluated with the displaceable water option toggled on, a **Displaceable Water** tab will be available (see Figure 93).

The Displaceable Water tab shows the following information about all water molecules available in the workspace:

- ID: Molecule ID (also shown in Properties Window).
- Type: [ignored], Displaced, or Non-Displaced. Indicates whether a water molecule is ignored (no interactions with ligand), displaced (because of non-favorable interactions), or non-displaced (favorable interactions with ligand).
- Detailed energy terms: Energy contribution, Energy (water-ligand), Energy (water-protein/cofactor), Energy (water-other waters).

From the **Options** check box it is possible to focus on displaced and non-displaced water molecules using the **Hide Ignored Waters** option or show all water molecules using the **Show All Waters** option.

From the list it is possible to visually inspect the water molecules in the 3D visualization window. By clicking on one or more entries in the list, the

corresponding water molecule is selected/highlighted in the 3D visualization window (if water molecules are toggled on in the Workspace Explorer window). The **Clear Selection** button can be used to clear all current selections.

The total energy contribution from the displaceable water interactions is the sum of all values in the **Energy contribution** column. This term, named

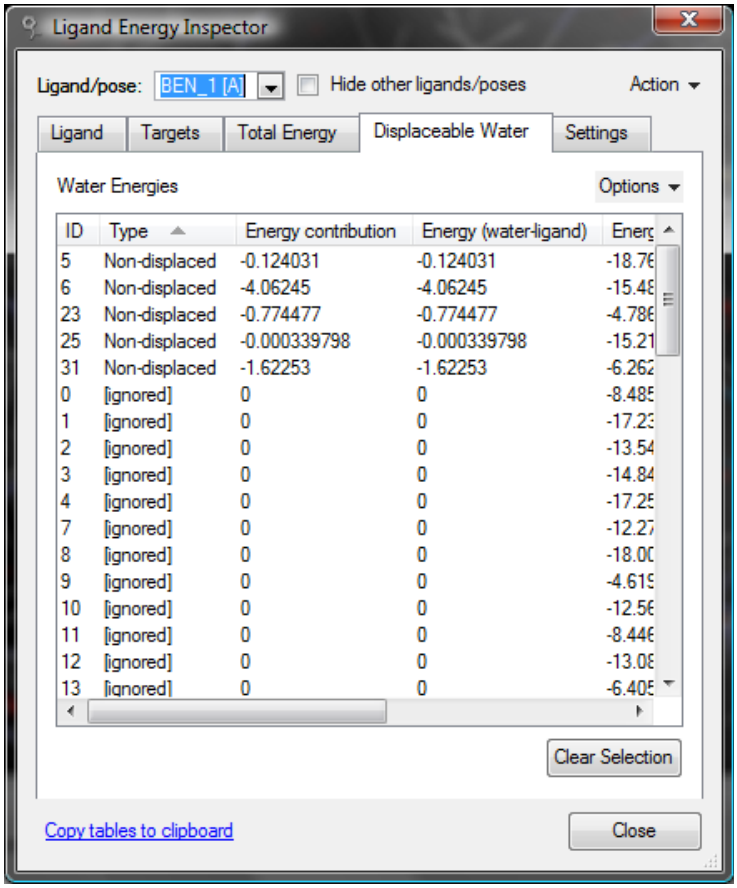


Figure 93: Displaceable Water tab: Listing of non-displaced, displaced, and ignored water molecules.

Displacable Water interactions, is also shown in the Energy Total tab (see Figure 94).

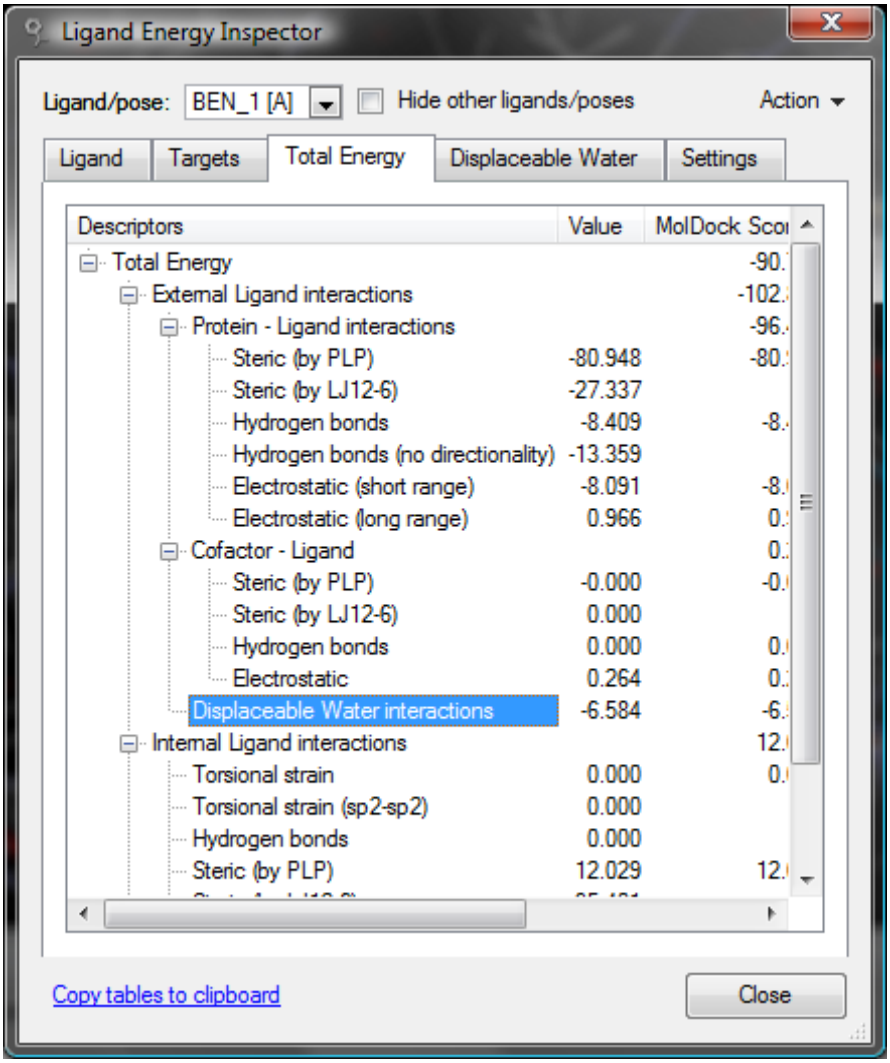


Figure 94: Energy contributions from various terms including the Displaceable Water Interactions.

For a more visual inspection of the displaced and non-displaced water molecules, it is possible to style the water atoms based on their individual energy contributions. This styling can be enabled by selecting the **Style Water Atoms by Energy** option from the **Action** menu: The radius of the water atoms will be scaled proportionally to their energy contributions and displaced waters are colored yellow, non-displaced waters are colored green if they are favorable and red if they are not favorable. Figure 95 shows an example of the **Style Waters by Energy** visualization style.

Notice that the styling is not updated automatically, so whenever a ligand is re-evaluated (using the **Re-evaluate** button located in the Settings tab), the action has to be selected again from the **Action** menu to update the visualization view.

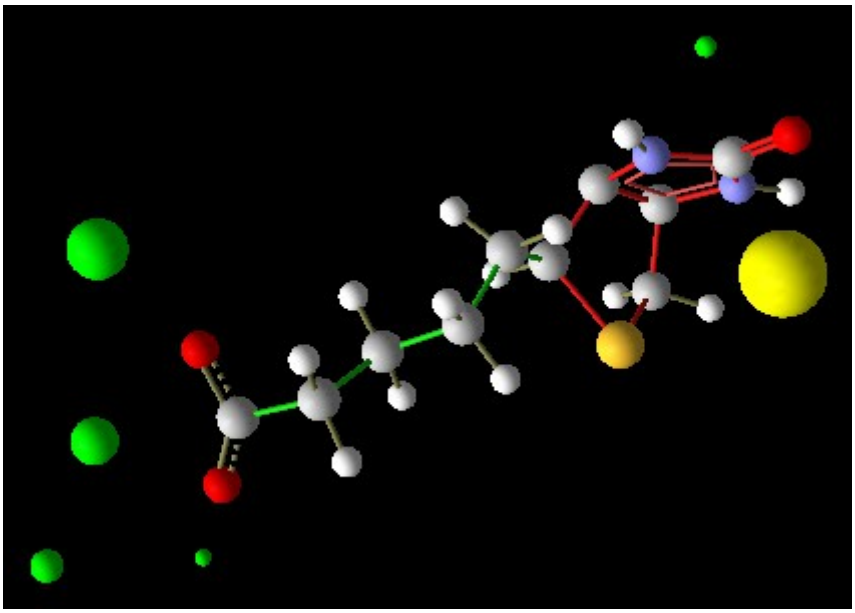


*Figure 95: Example of **Style Waters by Energy** action. The green spheres represent favorable non-displaced water molecules and the yellow sphere represents a displaced water molecule.*

10 Template Docking

Template docking can be used when knowledge about the 3D conformation of a ligand is available.

For instance, a protein might have one or more inhibitors with experimentally known 3D structures. From the known conformations it is possible to create a *template* with features expected to be relevant for the binding. This allows the docking engine to focus the search on poses similar to the docking template.

Docking templates can be used together with an ordinary docking scoring function (in order to focus or guide the search), but templates can also be used without any additional energy terms (for instance if no structural information about the target is known). This is useful for aligning ligands – by defining a template from one or more ligands as a reference template, and other molecules can then be docked and aligned to the template. Notice that template alignment takes the ligands flexibility into account: The docking engine will try to find the optimal conformation of the ligand when fitting to the template.

It is also possible to align molecules and extract detailed information about the similarity based on the overlap from each individual template point. This information can then be used in Molegro Data Modeller to create a regression model against some known empirical quantity (3D QSAR).

10.1 Template Scoring Function

Templates are implemented as scoring functions rewarding poses similar to the specified pattern.

A *template* is a collection of *template groups*, where each group represents a chemical feature for an atom (e.g. 'hydrogen acceptors atoms' form a template group). Each template group contains a number of *centers*: optimal 3D

positions for the group feature.

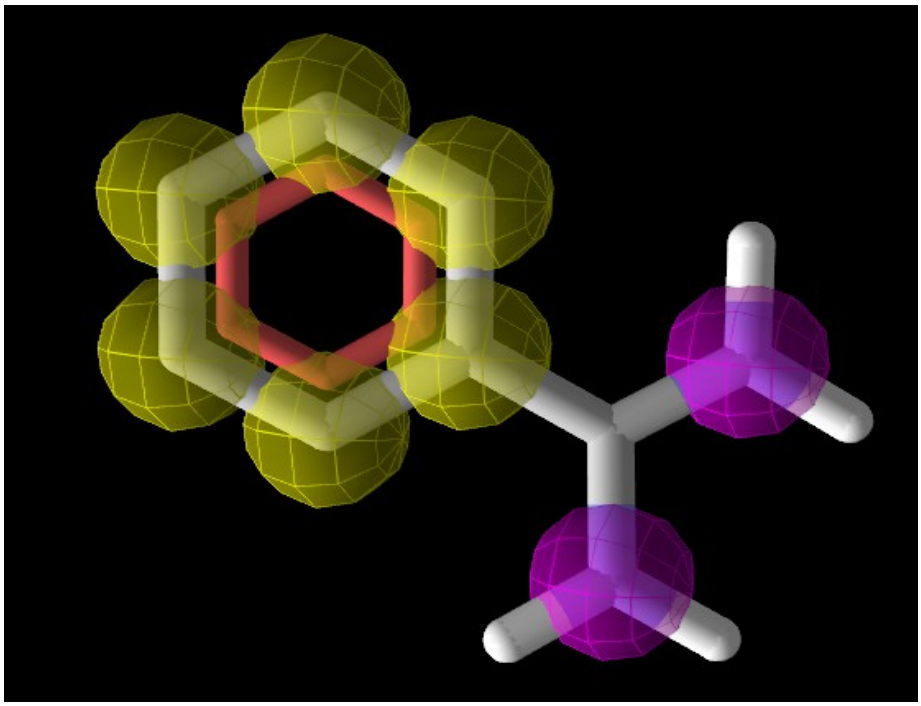


Figure 96: Example of a template with two groups: ring atoms (yellow) and hydrogen donor atoms (purple). The colored spheres indicates group centers.

If an atom matches a group definition (e.g. is a hydrogen acceptor), it will be rewarded depending on its distance to the group centers by using the following (Gaussian) formula for each center:

$$e = \omega * \exp(-d^2 / r_0^2)$$

where *d* is the distance from the position of the atom to the center in the group. ω is a weight (importance) factor for the template group, and r_0 is a distance parameter, specifying a characteristic distance for the template group (when *d* is equal to this characteristic distance, the interaction is at $e^{-1} \sim 36\%$ of its maximum value). ω and r_0 can be customized for each template group.

The following strategy applies when evaluating ligands during docking: For each atom in the ligand, score contributions from all centers in all matching groups are taken into account, i.e. a single atom may contribute to several centers in several groups - an atom is not restricted to the closest matching center or a single group.

The template score is normalized: the resulting score found using the procedure above is divided by the score of a perfectly fitting ligand (i.e. if the template was constructed from one ligand only this ligand would have a normalized template score of 1.0). Notice that in the docking wizard it is possible to specify an overall normalization of the similarity score term to

balance it with other scoring terms: the default overall normalization when docking is -500.0.

10.2 Setting up Template Docking

In order to setup template docking, import the desired reference ligands into the workspace and select **Docking** | **Setup template docking**.

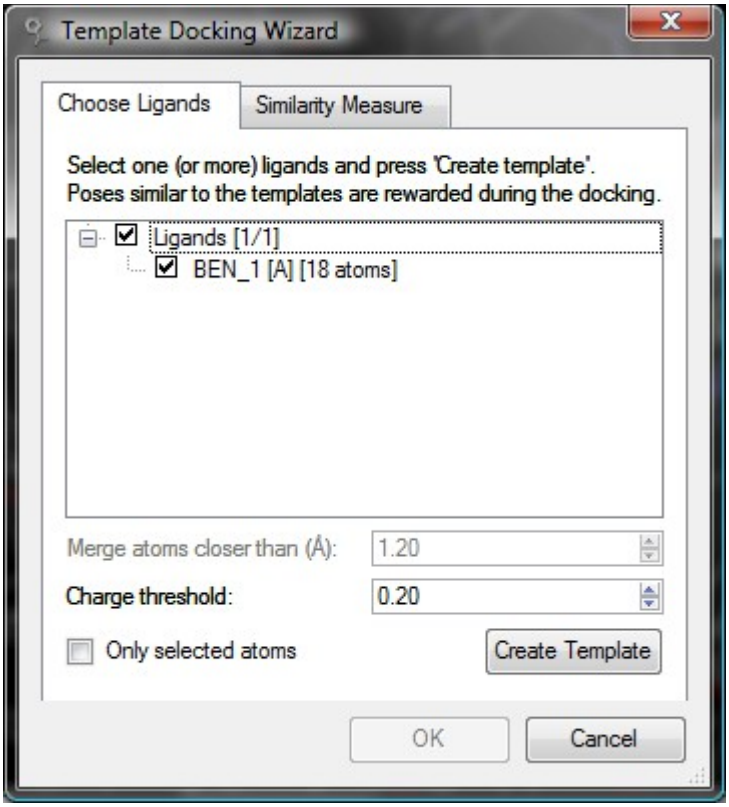


Figure 97: The Template Docking Wizard.

On the first tab in the template wizard, the reference ligands are specified.

When pressing the **Create Template** button, the docking template is created.

If only one ligand is selected, the procedure is straight forward: each atom in the chosen ligand is tested against the predefined template groups and if the atoms match, the position of the atom is added to the group as a new group center. Notice that only heavy atoms are taken into account when creating the template - hydrogen atoms are simply ignored. If **Only selected atoms** is checked, only the atoms that have been selected in the 3D view are taken into account – this can be useful for creating a template from a subset of a ligand.

If several ligands are chosen, MVD first creates a docking template from the first ligand as above. Then each atom from the remaining ligands are compared to the existing centers from the template being constructed. If an atom is closer to an existing center than the threshold specified in the wizard (default: 1.2 Å) the atom will be considered equal to that center. Notice that a

center can be part of several template groups: if any of the existing groups that the center is part of do not match the atom, the center is removed from them (the center is degraded in order to match both the current atom and the atom which defined the original group).

The **Charge threshold** option is used to specify a charge threshold (default: 0.2) for positive and negative charges. Atoms with a numerical charge less than this threshold are not considered charged.

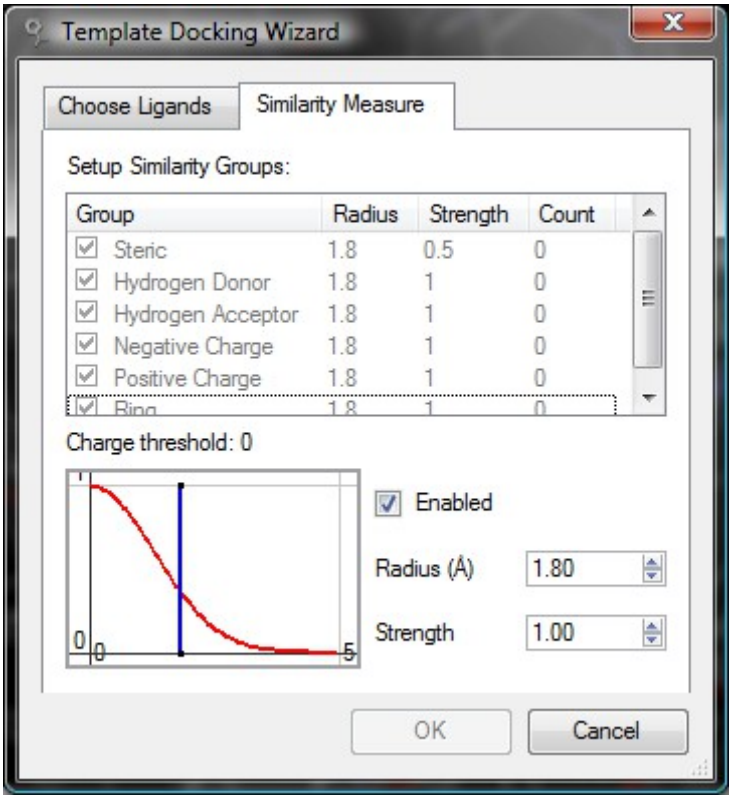


Figure 98: Customizing the similarity measure.

By choosing the **Similarity Measure** tab, it is possible to customize the similarity score. It is possible to enable or disable different template groups, and to adjust the Gaussian function used to compare the atom overlap with the group centers.

The following groups can be chosen:

- **Steric**. The steric group matches all atoms. It is used for shape matching without taking any chemical groups into account.
- **Hydrogen Donor**. Matches any hydrogen donor atom.
- **Hydrogen Acceptor**. Matches any hydrogen acceptor atom.
- **Negative Charge**. Matches negatively charged atoms. Notice that atoms with a numerical charge less than the specified 'Charge threshold' are not considered charged.

- **Positive Charge**. Similar to negative charge as described above but for positively charged atoms.
- **Ring**. Matches all atoms which are part of rings (both aromatic and aliphatic).

The list view shows the following information:

- **Radius**. The characteristic radius (r_0) for the template group (see 'Template Scoring Function' above).
- **Strength**. The strength (ω) or weight for the template group.
- **Count**. The number of centers in the group.

The different template groups will be visualized in the visualization window with a sphere for each center in the template group. Different template groups will be colored in different colors.

The small graph in the lower left corner shows the strength of the potential for the selected group as a function of radial distance. The vertical blue line indicates the characteristic radius, r_0.

Adjust the parameters as needed and press **OK** to add the template to the workspace.

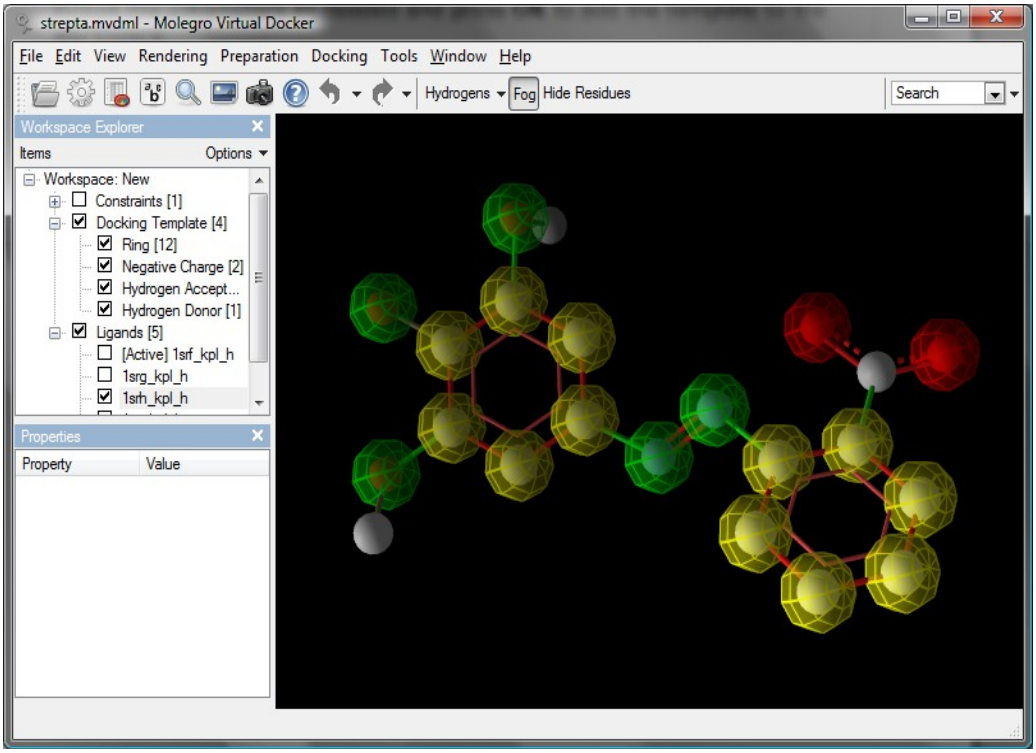


Figure 99: Visualization of template groups. Notice the corresponding categories in the workspace explorer.

When a docking template has been created, a new category **Docking Template** appears in the Workspace Explorer. The category can be expanded to reveal the different template groups it contains.

Using the context menu on the Docking Template category, it is possible to edit or remove an existing docking template. From the context menu it is also possible to choose **Open in Molegro Data Modeller**: this allows you to test each ligand or pose in the workspace against the template – the atom overlap for each template group center will be calculated and the resulting spreadsheet will be opened in Molegro Data Modeller. All values are normalized so a value of 1.0 corresponds to an optimal match. Each row in the spreadsheet corresponds to a ligand or pose, and each column corresponds to the overlap with a template group center. The columns are named S*x* for the steric group centers, HD*x* for hydrogen donors centers, HA*x* for hydrogen acceptors, and Pos*x*, Neg*x* and Ring*x* for the positive, negative and ring atom groups (where '*x*' is an index). Each group also has a sub-total match designated by an 'ALL' suffix (e.g. 'HD ALL').

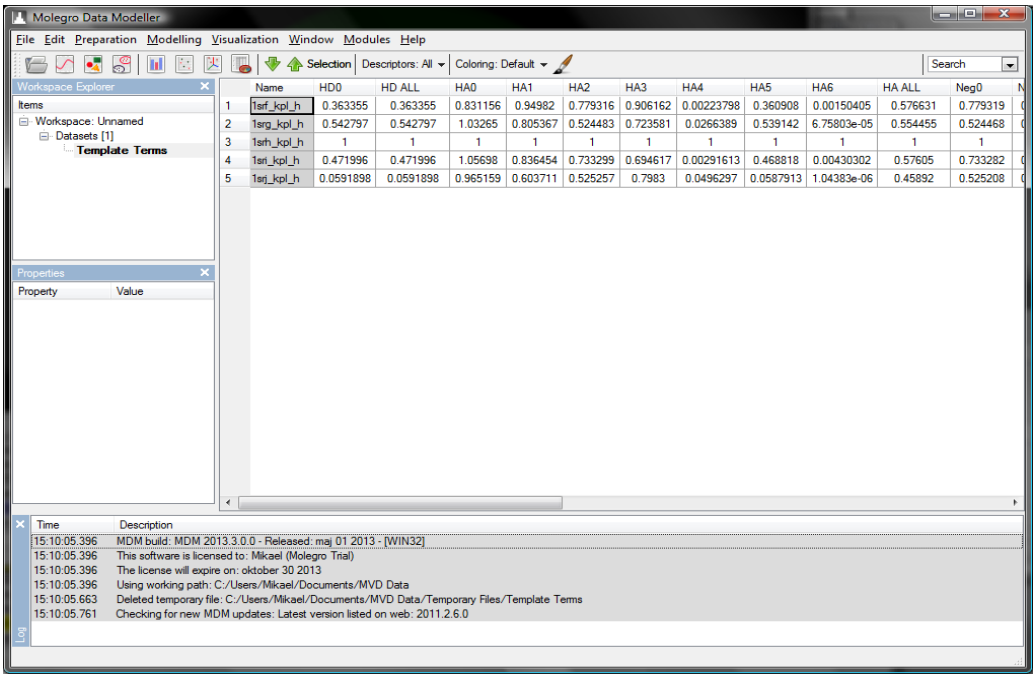

Molegro Data Modeller
File Edit Preparation Modelling Visualization Window Modules Help
Workspace Explorer
Items
Workspace: Unnamed
Datasets [1]
Template Terms
Properties
Property Value
Name HD0 HD ALL HA0 HA1 HA2 HA3 HA4 HA5 HA6 HA ALL Neg0
1 1srf_kpl_h 0.363355 0.363355 0.831156 0.94982 0.779316 0.906162 0.00223798 0.360908 0.00150405 0.576631 0.779319
2 1srg_kpl_h 0.542797 0.542797 1.03265 0.805367 0.524483 0.723581 0.0266389 0.539142 6.75803e-05 0.554455 0.524468
3 1srh_kpl_h 1 1 1 1 1 1 1 1 1 1 1
4 1sri_kpl_h 0.471996 0.471996 1.05698 0.836454 0.733299 0.694617 0.00291613 0.468818 0.00430302 0.57605 0.733282
5 1srj_kpl_h 0.0591898 0.0591898 0.965159 0.603711 0.525257 0.7983 0.0496297 0.0587913 1.04383e-06 0.45892 0.525208
Time Description
15:10:05.396 MDM build: MDM 2013.3.0.0 - Released: maj 01 2013 - [WIN32]
15:10:05.396 This software is licensed to: Mikael (Molegro Trial)
15:10:05.396 The license will expire on: oktober 30 2013
15:10:05.396 Using working path: C:/Users/Mikael/Documents/MVD Data
15:10:05.663 Deleted temporary file: C:/Users/Mikael/Documents/MVD Data/Temporary Files/Template Terms
15:10:05.761 Checking for new MDM updates: Latest version listed on web: 2011.2.6.0

Figure 100: Using Molegro Data Modeller for inspecting the docking template.

By analyzing a set of ligands aligned using template docking, it is possible to create a regression model of a experimentally known quantity. This would allow for a '3D QSAR' approach based on the values of the group center overlap.

10.3 Docking with Templates

Whenever a template definition is present in the workspace, the following tab appears in the Docking Wizard (after the first tab where the input ligands are chosen):

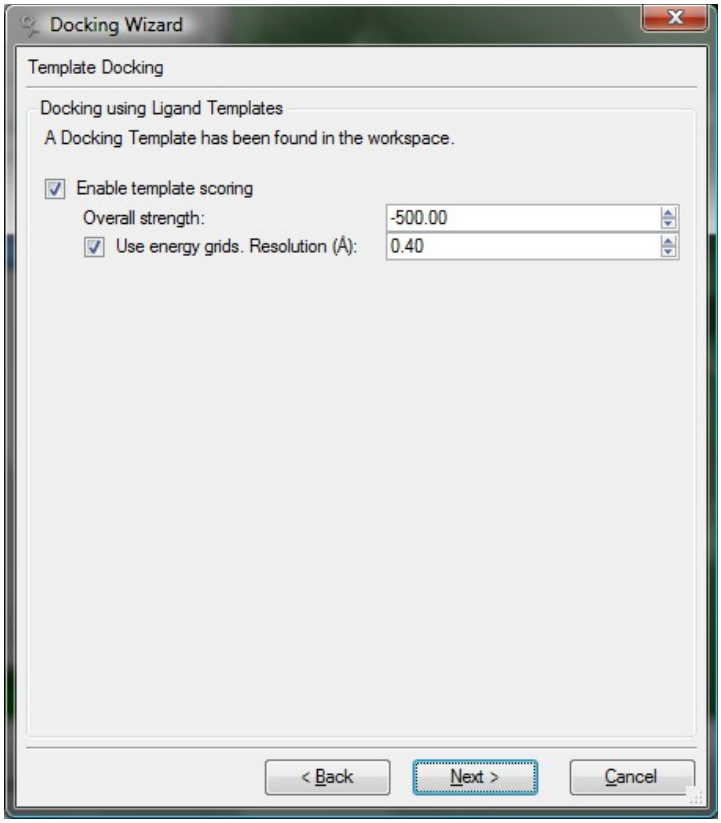


Figure 101: The Template Docking Tab in the Docking Wizard.

The **Overall strength** determines the normalization of the similarity score. A ligand perfectly matching the template gets an energy contribution corresponding to the specified strength (e.g. per default a perfectly matching ligand gets a energy contribution of -500). **Use energy grids** toggles whether grids with precalculated energy contributions should be used during the docking. It is recommended to use energy grids.

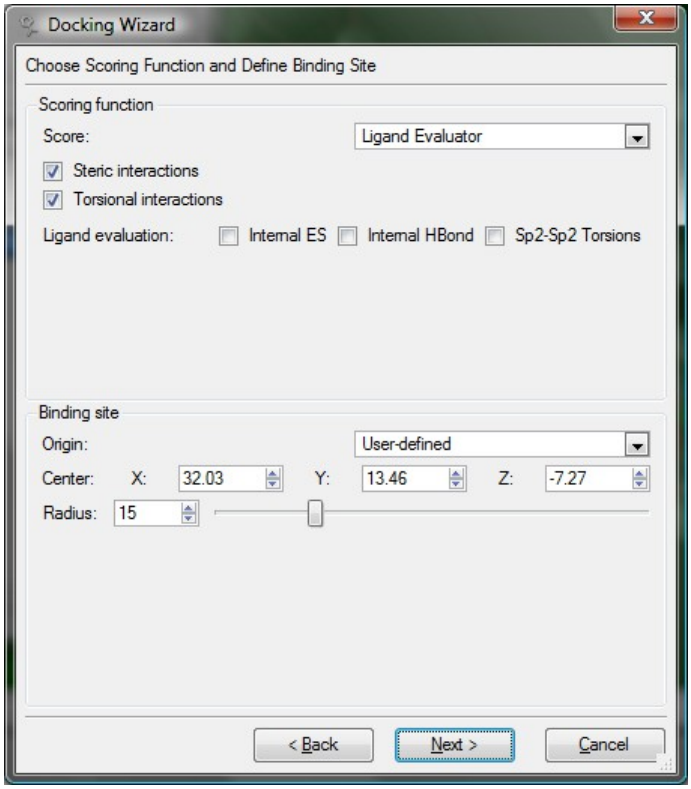


Figure 102: The 'Ligand Evaluator' scoring function.

When a similarity definition is present in the workspace, a new score function appears in the Docking Wizard: the 'Ligand Evaluator'. The Ligand Evaluator estimates the internal energy of a ligand and is identical to the E_{intra} term in Appendix I: MolDock Scoring Function. It is possible to enable or disable steric, torsional and electrostatic interactions.

When aligning molecules it is necessary to use the Ligand Evaluator to prevent internal collapse of the ligands – otherwise different atoms in the ligands might try to overlap each other in order to satisfy the same template group center. Notice that when docking against a protein target *combined* with a template, the Ligand Evaluator should not be used – choose a MolDockScore evaluator instead. The internal ligand energy terms in the MolDockScore will prevent the ligand from collapsing.

10.4 Inspecting Results

After the Docking engine has finished aligning or docking the ligands, the resulting poses are imported back into MVD in the same way as an ordinary docking result.

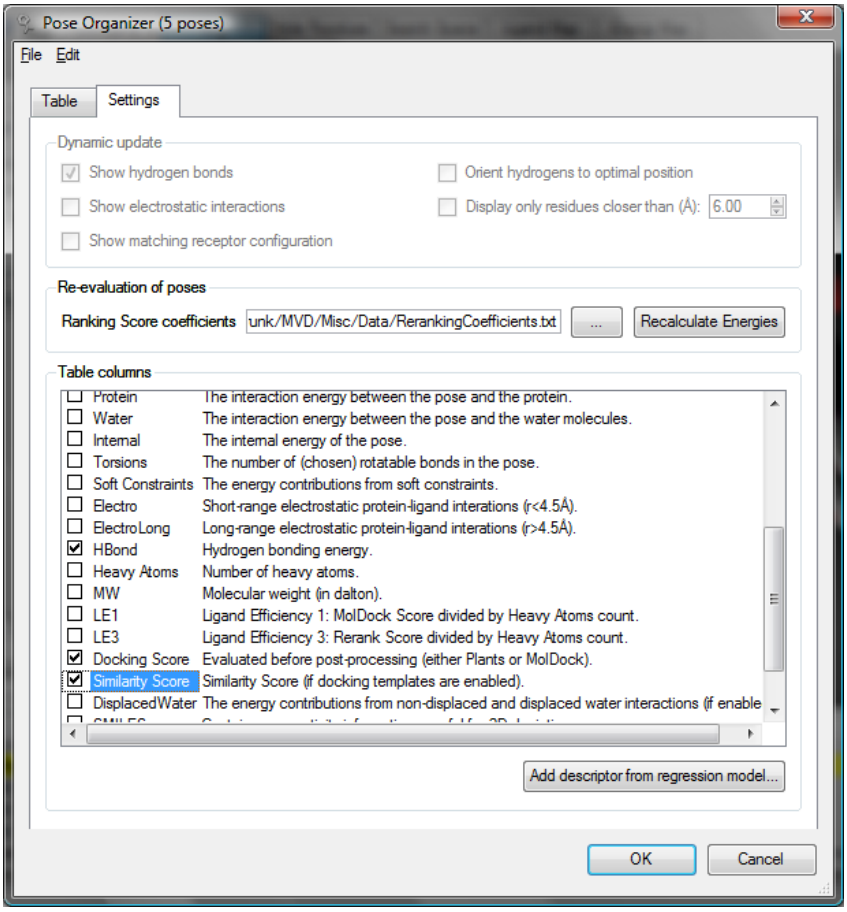

Pose Organizer (5 poses)
File Edit
Table Settings
Dynamic update
Show hydrogen bonds
Orient hydrogens to optimal position
Show electrostatic interactions
Display only residues closer than (Å): 6.00
Show matching receptor configuration
Re-evaluation of poses
Ranking Score coefficients unk/MVD/Misc/Data/RerankingCoefficients.txt ... Recalculate Energies
Table columns
Protein The interaction energy between the pose and the protein.
Water The interaction energy between the pose and the water molecules.
Internal The internal energy of the pose.
Torsions The number of (chosen) rotatable bonds in the pose.
Soft Constraints The energy contributions from soft constraints.
Electro Short-range electrostatic protein-ligand interations (r<4.5Å).
ElectroLong Long-range electrostatic protein-ligand interations (r>4.5Å).
HBond Hydrogen bonding energy.
Heavy Atoms Number of heavy atoms.
MW Molecular weight (in dalton).
LE1 Ligand Efficiency 1: MolDock Score divided by Heavy Atoms count.
LE3 Ligand Efficiency 3: Rerank Score divided by Heavy Atoms count.
Docking Score Evaluated before post-processing (either Plants or MolDock).
Similarity Score Similarity Score (if docking templates are enabled).
DisplacedWater The energy contributions from non-displaced and displaced water interactions (if enable
Add descriptor from regression model...
OK
Cancel

Figure 103: Enabling the Similarity Score term.

It is important to notice that the Pose Organizer table only shows the contributions from the primary score function (the Ligand Evaluator or the MolDock Score funtion). The similiary contribution from the docking template is not shown per default. In order to see the similiary score go to the **Settings** and enable **Similarity Score** from the list of table columns. To see the score actually assigned to the pose during docking, enable **Docking Score** – this will be the sum of the similarity score and the chosen primary score function.

11 Customizing Molegro Virtual Docker

11.1 General Preferences

Molegro Virtual Docker can be customized using the **Preferences** dialog, which can be invoked from the **Edit** menu or by pressing **F4**. Preference settings are categorized in **General**, **Graphics**, **Mouse**, and **Parsing** tabs.

In the **General** tab (see Figure 104), the following settings are available:

- The **Load most recent workspace on startup (if any)** option toggles automatic import of the last used workspace.
- The **Show tip of the day on startup** option toggles whether the **Tip of the day** dialog box is shown during startup or not.
- The **Check for new updates on startup** option enables MVD to automatically check for new updates during startup.
- The **Create system log (in directory below)** option is used to toggle whether a system log is created for each execution of MVD. The system log contains information about user actions conducted and is used to track potential bugs and performance problems. By default, the log files are stored in the `Logs` directory located in the same directory as the `mvd` executable file but another directory can be used if needed (e.g. if user has no write permissions to the directory used). *Notice:* If you encounter problems with MVD please email the log file created before the crash to: support@clcbio.com
- The **Working directory** setting is used to set the current **Working directory**, which is the root path for file related operators (e.g. when loading and saving molecular structure files and log files).
- The **Virtual Grid executable** and **PDF viewer** settings are used to

specify the location of the executable files for Molegro Virtual Grid and a PDF viewer for reading the user manual. The default PDF viewer specified by the operating system will be used if no executable file is provided.

- The **CUDA device** setting is used to specify default CUDA device ID. See Section 6.5 for more details.
- The **Level of details for docking file logs** option is used to specify the level (amount) of information that will be saved to the time-stamped log files created during the docking simulations. In particular, the None and Errors options are suitable for virtual screening runs since the amount of information saved will be small. The setting is used for all docking runs started on the local machine where MVD is installed.

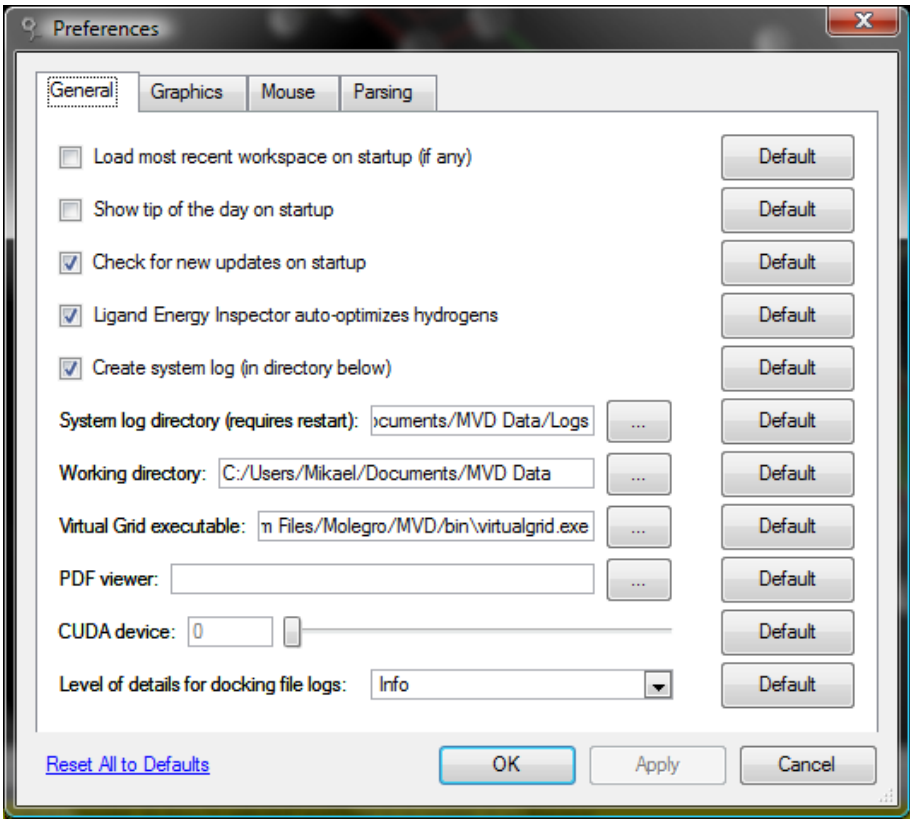


Figure 104: General preference settings.

The **Graphics** tab (see Figure 105) contains settings related to the **Visualization Window**:

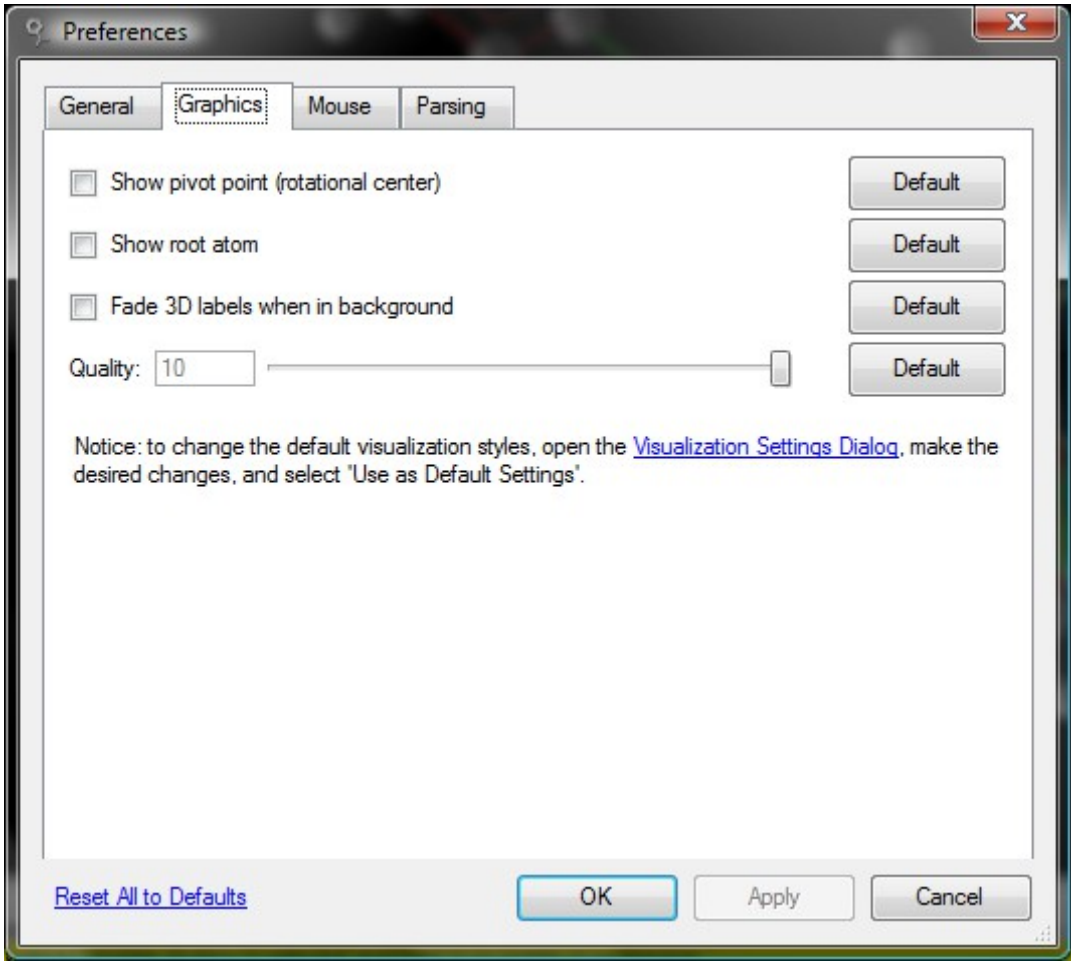


Figure 105: The graphics tab of the Preferences dialog.

- The **Show pivot point (rotational center)** option toggles the visibility of the pivot point (small grayish ball).
- The **Show root atom** option toggles the visibility of the currently chosen root atom for each of the ligands in the workspace (see 'Set root atom' in Section 4.3 for more info).
- The **Fade 3D labels when in background** option toggles fading of labels in the **Visualization Window.**
- The overall rendering quality can be specified using the **Quality** option. Modern computers with dedicated 3D hardware should be able to run at highest quality even when rendering relatively large molecules. It is easy to test new quality settings by selecting the level of quality and pressing the **Apply** button.

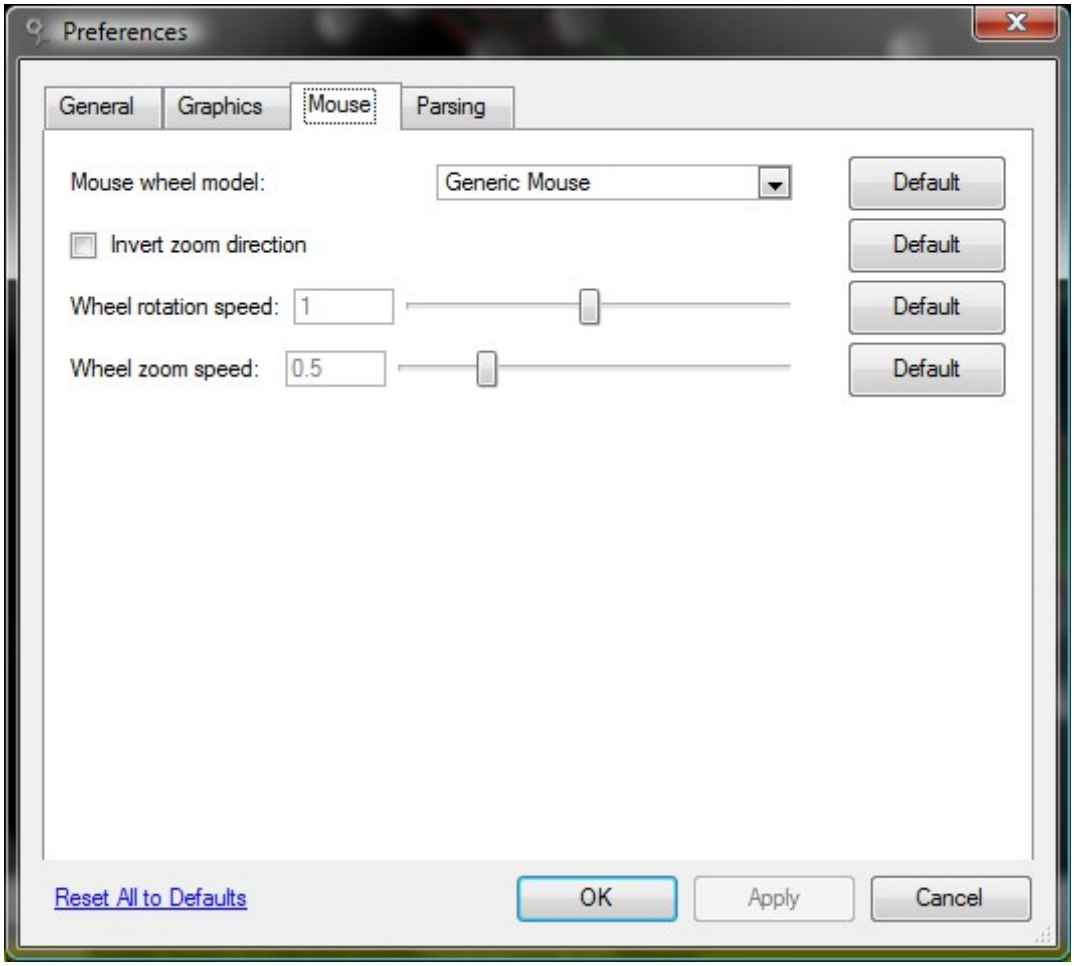


Figure 106: Mouse Preferences.

The **Mouse** tab customizes how the mouse interacts with the 3D world. MVD supports the 360 degrees scroll-ball on the *Apple Mighty Mouse*. Currently, the 360 degrees scroll-bar feature is only supported on Mac OS X (since no mouse drivers are available for other platforms), but the mouse still works as a generic mouse on Windows and Linux.

To enable *Apple Mighty Mouse* support select it under **Mouse wheel model**. When **Apple Mighty Mouse** mode is selected, the scroll-ball can be used to rotate the 3D world. Additionally, the scroll-ball button can be used to zoom in the 3D world by pressing the button while using the scroll-ball as a standard mouse-wheel. However, to enable the zoom option, the scroll-ball button should be set to **Button 3** in the *Mac OS X Mouse preferences* dialog (see Figure 107).

Invert zoom direction toggles how the 3D worlds zooms – rotating the scroll wheel towards the user will normally make the 3D objects appear larger, but this behavior can be inverted by toggling this option on. The setting also applies to zooming using both mouse buttons.

It is also possible to adjust the mouse wheel sensitivity (by using the **Wheel rotation speed** and **Wheel zoom speed** sliders).

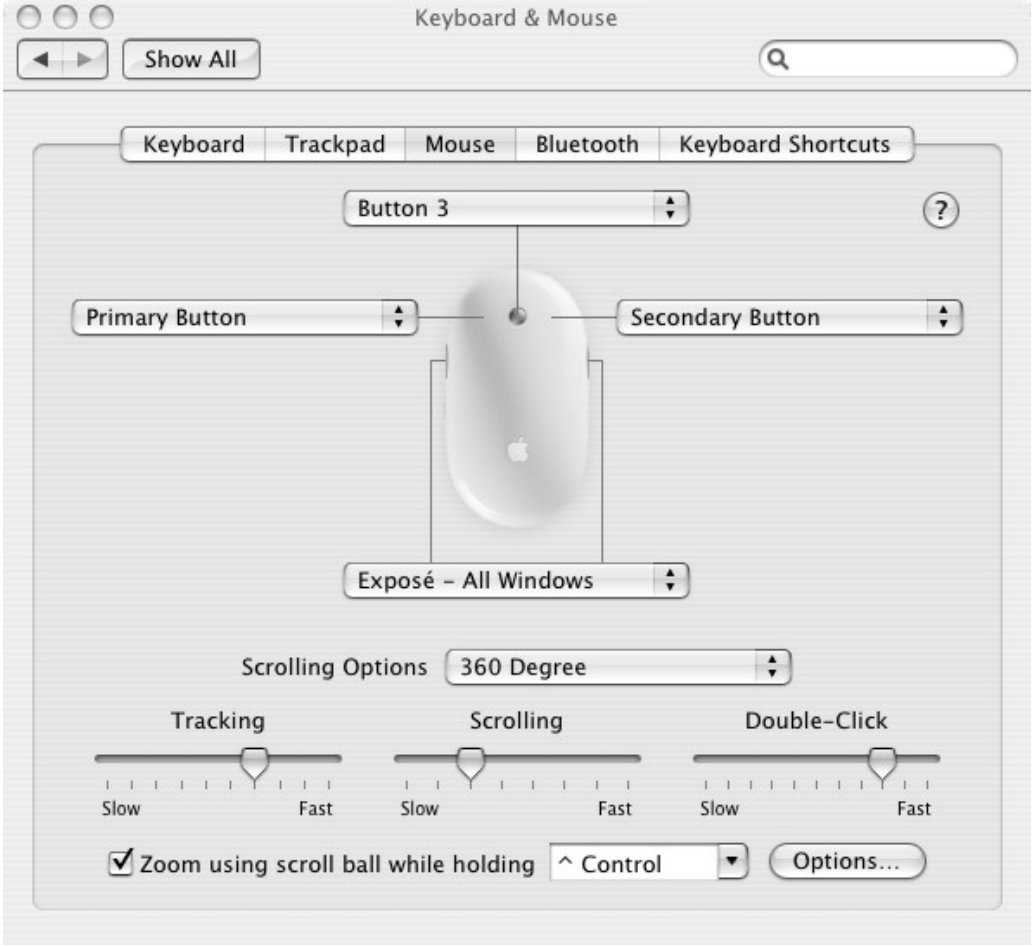


Figure 107: Mighty Mouse preferences on Mac OS X.

The final settings tab, **Parsing**, contains the **Minimum protein size (PDB import)** option. This option is used for setting the minimum number of heavy atoms required for parsing a molecule as a protein during PDB import (default is 69 heavy atoms). If the parsed molecule contains less heavy atoms than the specified threshold value it is parsed as a ligand (and residue information is ignored).

The **Parsing** tab also determines how MVD handles non-standard characters (such as special national characters). This setting is used when importing and exporting molecular structures in text file format (such as SDF, Mol2, PDB files) and when working with other text files (such as 'mvdresults' and 'mvdscript'). XML files (such as MVDs internal MVDML file format) are always stored as UTF-8. Notice that the Batch Job Script Parser always uses UTF-8 as default encoding (it runs in another process and is not aware of the MVD settings).

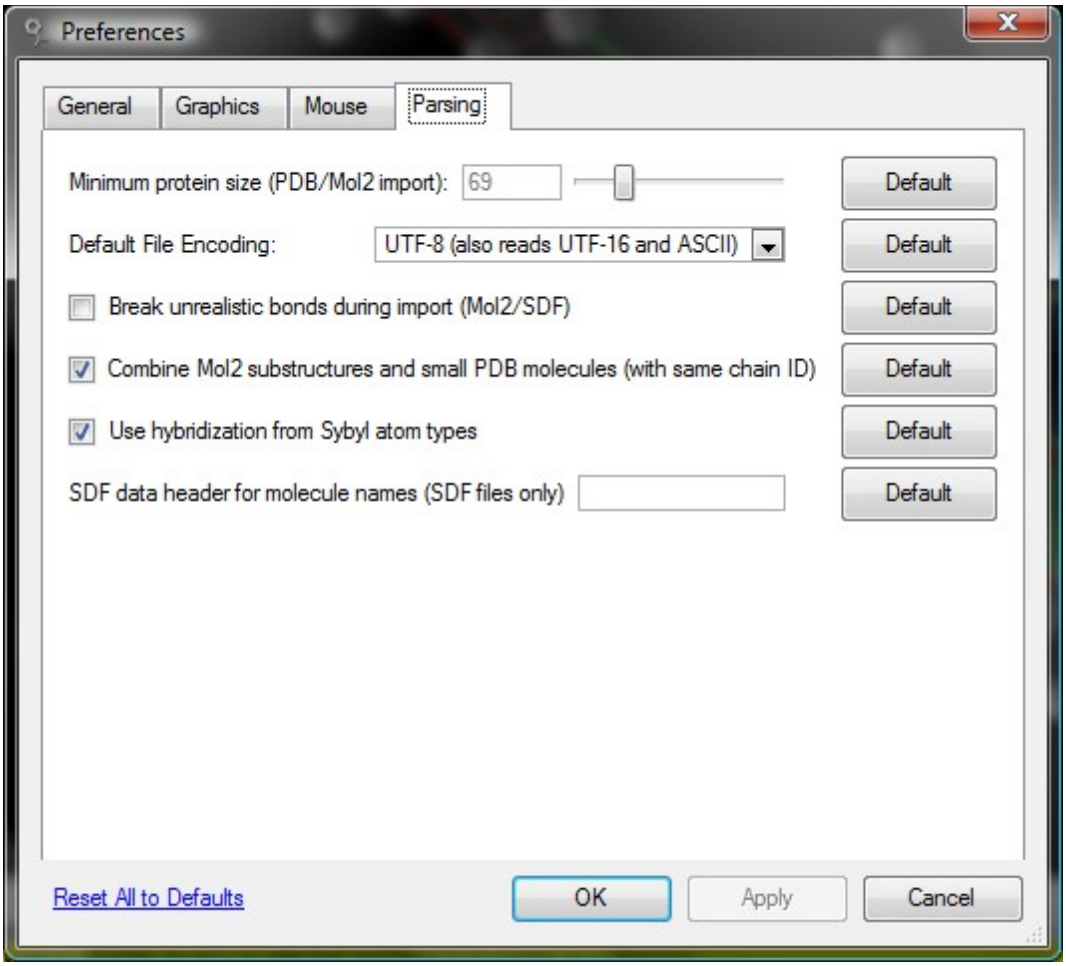


Figure 108: Parsing preferences.

The **Default File Encoding** drop-down box allows you to choose which encoding should be used. It is recommended to use the default setting, UTF-8 Unicode. Using the UTF-8 encoding all Unicode characters can be encoded and since molecular data files rarely contain special characters, it is more space-efficient than UTF-16 (where each character always uses at least 2 bytes). Files stored as 8-bit ANSI/ASCII files will also be imported correctly as Unicode if they do not contain any special national characters, and UTF-16 will also be automatically recognized in this mode. It is also possible to store data as Locale 8-bit. In this encoding all characters are stored as a single byte, meaning only 256 characters can be represented. The actual characters included in this set depends on the current national codepage settings on the machine. This option should only be used when exporting data to older software products not capable of parsing Unicode text.

Break unrealistic bonds during import (Mol2/SDF) determines whether or not unrealistic bonds parsed from Mol2 or SDF files should be ignored during import. A bond is considered *unrealistic* if the distance between two bonded atoms is more than the sum of their covalent radii plus a threshold of 0.7Å.

The **Combine Mol2 substructures and small PDB molecules (with same chain ID)** option is used to decide whether or not molecule fragments should be combined during import. Molecule fragments can be combined if any atom in one fragment can form a covalent bond to any other atom in another fragment. Molecule fragments can only be combined if they share either Mol2 substructure IDs or chain IDs in the case of PDB files.

When the **Use hybridization from Sybyl atom types** option is enabled, Sybyl atom types will be used to determine hybridization (if they are available during import). Otherwise, the default geometric heuristic is used (see Appendix VII: Automatic Preparation for details).

The final option, **SDF data header for molecule names (SDF files only)**, can be used to specify the name of the SDF data header that will be used for naming molecules during import instead of using the first line in each molecule header. The first line will also be used if the file does not contain the specified data header.

The preference settings are stored when exiting the MVD application. The location of the saved settings depends on the operating system used:

- Windows: the settings are stored in the `system registry`.
- Mac OS X: the settings are stored in a `com.molegro.MVD.plist` file located in the `<user folder>/Library/Preferences/` folder.
- Linux: the settings are stored in a `mvdrc` file located in a hidden folder named `<user folder>/.molegro`.

11.2 Command Line Parameters

Currently, the following command line parameters are available:

```
<filename>
-nogui
-interactive
-currentPath
-cudadevice <ID>
-licensedir
-macro=<label>
```

The `<filename>` parameter can be used to import molecular files during MVD startup. If more than one file is listed (separated by spaces), each file will be imported.

Example: `/Molegro/MVD/bin/mvd 1stp.pdb`

If the filename has `mvdscript` as file extension (e.g. `mydocking.mvdscript`), a script parsing progress dialog will be invoked and the script will be parsed and interpreted.

The `-nogui` parameter can be used to run the script job without invoking the progress dialog.

Example: `/Molegro/MVD/bin/mvd mydocking.mvdscript -nogui`

Using the `-interactive` parameter, MVD can be started in interactive mode which is used to allow scripting languages (e.g. Python) to interact with MVD and control the docking process. See Chapter 17 for more details.

The `-currentPath` parameter can be used to override the working directory specified in the general preference settings with the current path. This is particularly useful when running MVD from different working directories (using a terminal window) or when using a script to start up MVD.

Example: `/Molegro/MVD/bin/mvd -currentPath`

The `-cudadevice <ID>` parameter can be used to specify the CUDA device ID from the command line.

Example: `/Molegro/MVD/bin/mvd -cudadevice 0`

The `-licensedir` parameter can be used to specify another directory where the MVD license is located. By default, MVD checks for the license file in the same directory as the MVD executable (e.g. Molegro/MVD/bin).

Example: `/Molegro/MVD/bin/mvd -licensedir /Molegro/License`

Finally, the `-macro=<label>` parameter can be used to specify a macro that is executed when starting up MVD. This can be useful for e.g. setting up a user-customized visualization style when running MVD. The macro label is used to identify which macro to execute (the labels can be added or modified in the Macro and Menu Editor dialog). Notice that labels are not allowed to contain white spaces.

`Example: /Molegro/MVD/bin/mvd -macro=MyOwnMacro`

11.3 Changing Re-ranking Score Coefficients

The energy terms and their weights (coefficients) used in the reranking scoring function can be altered by modifying the `RerankingCoefficients.txt` file located in the `/Misc/Data/` directory (located within the main directory of MVD).

Notice: Changing these coefficients and disabling/enabling energy terms will alter the performance of the reranking score used in the **Pose Organizer** dialog and may result in much worse performance. Remember to backup the original file before modifying the coefficients.

12 Obtaining the Best Docking Results

This section takes a closer look at the most important aspects regarding preparation, docking, and post-analysis that can be decisive for whether docking with Molegro Virtual Docker will be successful or not. By taking the following suggestions into account, we hope that common pitfalls can be avoided.

12.1 Preparation

- General issues: It is recommended to remove unwanted material such as proteins, ligands, cofactors, and water molecules if they are not needed in the actual docking simulation.
- Validation: The automatic preparation of molecules might fail in some cases. It is therefore advisable to manually inspect the molecules (in particular ligands) and check bond orders, hybridization states, and if hydrogens are correctly assigned.
- Protonation: If the protein is expected to have unusual protonation states near the binding site, be sure to set them using the **Protein Preparation** dialog.
- Ligand flexibility: By default, all torsions in the ligand that can be flexible are set flexible during the docking simulation. The complexity of the docking search can be significantly reduced, if the number of torsions that are set flexible during the docking run is lowered. Bonds can be set rigid during docking using the context menu (right-click on the bond and select **Set Flexibility** | **Rigid while docking**).
- Cavity detection: Before starting the docking run, all potential binding sites (active sites) should be identified using the **Detect Cavities** dialog. The default settings listed in the wizard are generally applicable.

However, for large proteins or proteins having a lot of cavities, it is sometimes necessary to increase the number of cavities reported (**Max number of cavities**). Also remember to set the binding site **Origin** (in the **Docking Wizard**) to the specific cavity being investigated.

- Domain knowledge: The success of the docking run can be significantly improved if any domain knowledge is available. For instance, knowledge about preferred binding mode or ligand conformation can be used to set constraints or reduce the search space covered (e.g. constraints and binding site settings in the **Docking Wizard**).
- In some cases, docking performance can be improved by selecting another ligand root atom (right-click on ligand atom and select **Set as Root Atom**. The current root atom can be visually identified if visualization of root-atoms is enabled (see Section 11.1). The root atom is used as root in the torsion tree that is constructed when docking flexible ligands. Docking performance may be improved by setting the root atom in a region of the ligand that is suspected to contribute significantly to the docking energy.

12.2 Docking

- Size of search space: The size and location of the volume that the docking search algorithm will sample is defined by the **Binding site** settings in the **Docking Wizard**. Before starting the docking run, potential cavities should be identified (see Section 6.1). Found cavities can be used to specify the origin of the search space (in the **Docking Wizard**) and constrain candidate solutions to the region covered by the cavity (by enabling the **Constrain poses to cavity** option in the **Docking Wizard**).

  Notice: It is important to select a search space **Radius** that allows the ligand to be positioned within the search space region (typically between 15 and 20 Å). However, the **Radius** should be set as small as possible to make the docking search efficient. Likewise, the **Origin** (center) of the search space region can be manually adjusted to focus the sampling of candidate solutions to a specific region. This is particularly important if the cavity volume is much bigger than the ligand (for large cavities, focusing on one specific part of the cavity will significantly increase the docking accuracy).
- Search parameters: The default settings for the docking search algorithm are generally applicable. However, in some cases (e.g. for ligands with more than 15 torsions) it can be advantageous to increase the **Population size** to 100 individuals or more.
- Multiple runs: Because of the stochastic nature of the docking search algorithm, it is recommended to make multiple runs for each ligand-

protein setup. Typically, about 5-10 runs are needed to ensure convergence to the lowest-energy solution. For large ligands with more than 10-15 flexible bonds, 20-50 runs are sometimes needed. Additionally, it is recommended to cluster the returned poses (see Section) to lower the number of similar poses reported when taking all docking runs into account.

- Multiple poses: It is advisable to return multiple poses for each docking run (typically between 3 and 10) and rerank the poses found afterwards (see **Ranking poses** bullet below).
- Check warnings: The last tab in the **Docking Wizard** highlights potential warnings and errors. It is important to inspect the warning messages and see if further actions are needed. Otherwise, the docking run might be unsuccessful.

12.3 Post-analysis

- Ranking poses: The most promising poses returned when the docking run terminates can be further analyzed in the **Pose Organizer**. Ideally, the highest-scoring pose should represent the best-found binding mode. However, this is not always the case. A useful feature is to evaluate the poses using the **Reranking Score**. The **Reranking Score** makes use of a more advanced scoring scheme than the docking scoring function used during the docking run. Using the Reranking Score will often increase the accuracy of the ranked order of the poses.

13 Molegro Data Modeller Integration

Molegro Data Modeller is a powerful tool for data mining, data modelling, and data visualization.

Since Molegro Virtual Docker 6.0, Molegro Data Modeller is available as an integrated standard component, and replaces the previous Data Analyzer. It can be launched from inside Molegro Virtual Docker using **Tools** | **Molegro Data Modeller**. There are also short-cuts for viewing data in Molegro Data Modeller when working with the Pose Organizer (see page 111), Molecular Descriptors (see page 174), and when doing advanced template regression models (see page 153).

Molegro Data Modeller can be used for:

- Regression: Multiple Linear Regression, Partial Least Squares, Support Vector Machines, and Neural Networks.
- Classification: K-Nearest-Neighbors and Support Vector Machines.
- Chemistry: 2D depictions in spreadsheets and plots. SDF and SMILES support.
- Feature selection and cross-validation is simple to set up and use (using built-in wizards).
- Automated fine-tuning of regression model parameters (using grid-based search).
- Principal Component Analysis (PCA).
- Visualization: Histograms, 2D scatter plots, 3D plots, and Spring-Mass Maps.
- Clustering: K-means clustering and density-based clustering.
- Outlier Detection.
- Similarity Browser.

- Sophisticated subset creation: create diverse subsets by sampling from n-dimensional grids.

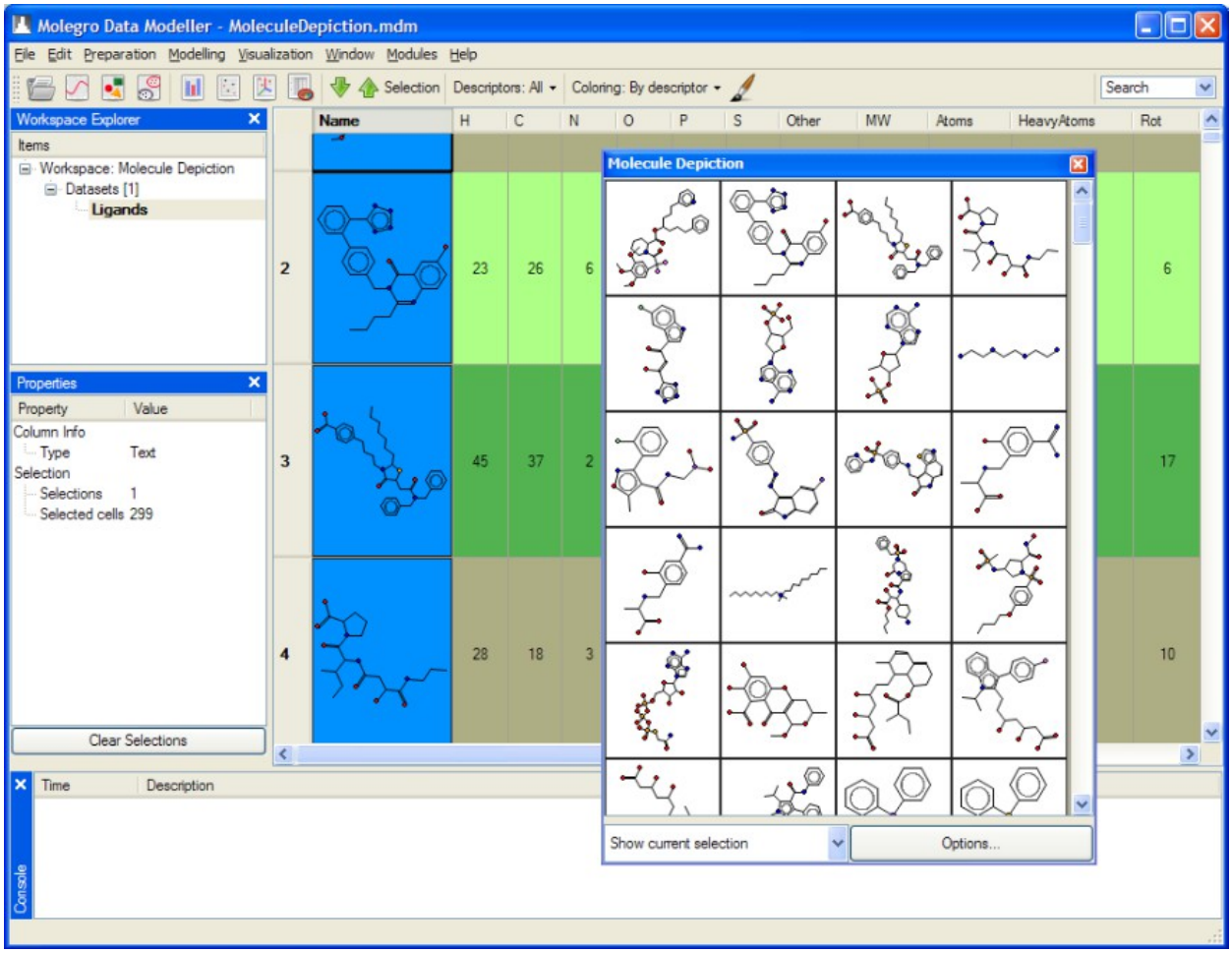


Figure 109: User Interface of Molegro Data Modeller

Molegro Data Modeller comes with its own separate manual. The manual may be accessed using the application menu in either Molegro Virtual Docker or Molegro Data Modeller using **Help** | **Molegro Data Modeller Manual**. The manual is also available in the 'Help' folder, in the installation directory.

Molegro Data Modeller is not restricted to working with Molegro Virtual Docker, and can be used with many kinds of data. For instance, Molegro Data Modeller may be used to build QSAR models for compounds without a 3D structure.

It is possible to run Molegro Data Modeller as a stand-alone application, by invoking the executable ('mdm.exe' or 'mdm') directly from the installation directory or using the desktop or start menu shortcuts (only available on Windows).

14 Molecular Descriptor Calculations

The **Descriptor Calculation Wizard** offers an interface for calculating molecular descriptors for small molecules.

Molecular descriptors are sets of numbers which quantifies and characterizes certain characteristics of a molecule. Several classes of molecular descriptors exists. For instance *molecular weight* is a molecular descriptor which only depends on the molecular formula for a given molecule, but descriptors may also rely on connectivity information (2D or topological descriptors) or may rely on the actual 3D conformation of the molecule (3D descriptors).

Molecular descriptors are typically fast to calculate – for instance the topological CFDM descriptors (described later) can be calculated for more than 1000 compounds per minute. This makes molecular descriptors very useful for e.g. initial filtering or clustering of a molecule library.

MVD is able to calculate molecular descriptors for all types of structural files that can be imported into the GUI or read from a Data Source (e.g., PDB, Mol2, MVDML, SDF, ...). As of now MVD does not parse SMILES strings or other 2D representations of molecules, even though the molecular descriptors in MVD are dependent only on the 2D properties of the molecule.

14.1 Using the Descriptor Calculation Wizard

The descriptor calculation wizard can be invoked from the main menu, by choosing **Tools** | **Descriptors Calculation Wizard**.

The first step is to specify which molecules the descriptors should be calculated for. This selection interface is identical to the one in the Docking Wizard. It is possible to calculate descriptors for molecules in the current workspace or from a chosen Data Source.

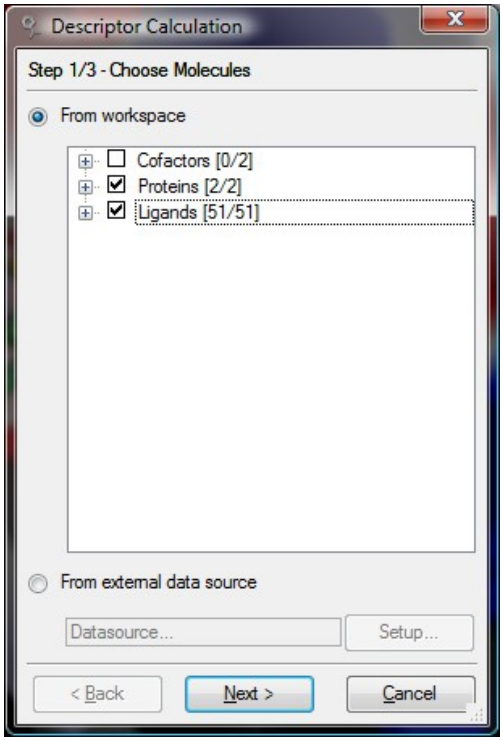


Figure 110: Choosing molecules in the Descriptor Calculation Wizard.

When calculating molecular descriptors for a large set of molecules, it is always advisable to use the data source import, since importing large molecule libraries into the graphical user interface requires all molecules to be present in memory at once and will slow the system.

Most of the molecular descriptors in MVD can only be calculated for small molecules, and will automatically skip the calculation for proteins. Also notice that when importing molecules from an external data source in PDB format, only the ligands in the file are imported – protein, cofactors, and water molecules are ignores.

14.2 Descriptors in MVD

The next step is to choose which descriptors to calculate.

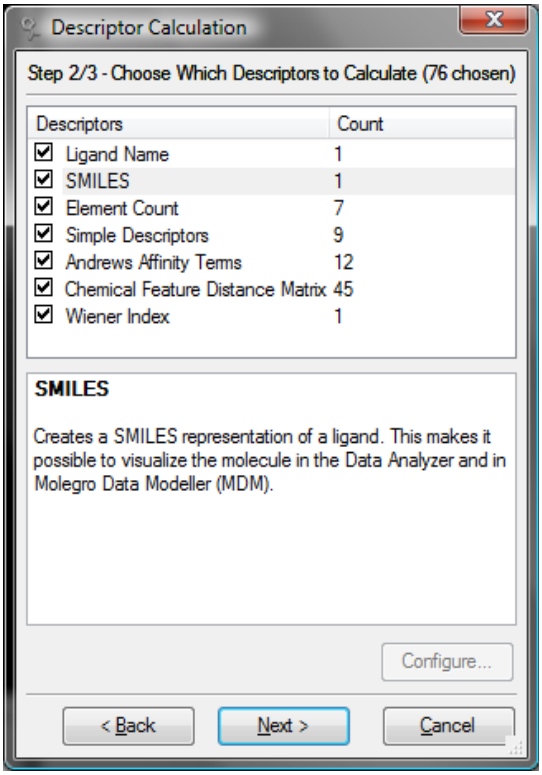

Descriptor Calculation
Step 2/3 - Choose Which Descriptors to Calculate (76 chosen)
Descriptors
Count
Ligand Name 1
SMILES 1
Element Count 7
Simple Descriptors 9
Andrews Affinity Terms 12
Chemical Feature Distance Matrix 45
Wiener Index 1
SMILES
Creates a SMILES representation of a ligand. This makes it possible to visualize the molecule in the Data Analyzer and in Molegro Data Modeller (MDM).
Configure...
< Back
Next >
Cancel

Figure 111: Choosing descriptors in the Descriptor Calculation Wizard.

The following categories of descriptors are available:

Category	Details
Ligand Name	Not a numerical descriptor, simply adds a column with the name of the compound to the output.
SMILES	Creates a SMILES string suitable for 2D representations. SMILES strings can be visualized as 2D molecule depictions in Molegro Data Modeller.
Element Count	Counts the number of atoms for a given element. By default H,C,N,O,P, and S are counted. All other elements are counted as 'other'. The elements may be customized using the **Configure...** button.
Simple Descriptors	A set of common descriptors including molecular weight, hydrogen donor / acceptor count, and other simple descriptors. The available descriptors are:

	MW - Molecular Weight **Atoms** - Atom count (including hydrogens) **HeavyAtoms** - Atom count (excluding hydrogens) **Rot** - The number of rotatable bonds **Rot2** - The number of rotatable bonds – but excluding any bonds which only rotates terminal hydrogen atoms. **HD** - The number of hydrogen donors **HA** - The number of hydrogen acceptors **Rings** - The number of rings **Aro** - The number of aromatic rings
Andrews Affinity Terms	An Andrews Affinity measure together with the terms needed for the calculation. These terms are described in: 'Functional goup contributions to drug-receptor interactions' PR Andrews, DJ Craik, JL Martin Journal of medicinal chemistry 27:1212, 1648-1657, American Chemical Society, 1984.
Chemical Feature Distance Matrix	The CFDM descriptors were created by Molegro and are described in details in the last section ('Chemical Feature Distance Matrix Descriptors') of this chapter. The CFDM descriptors are obtained by calculating the minimum, maximum, and mean topological distance between all pairs of chemical features. The topological distance is defined as the smallest number of covalent bonds between the two features. The following chemical features are investigated: hydrogen acceptors, hydrogen donors, positively and negatively charged atoms, and ring systems. Notice that a minimum charge of ±0.2 is required for an atom to be considered charged (this threshold may be changed in the settings dialog).
Wiener Index	The Wiener Index is the sum of the topological distance between all heavy atom pairs.

14.3 Choosing an Output Format

The final step is to choose an output format.

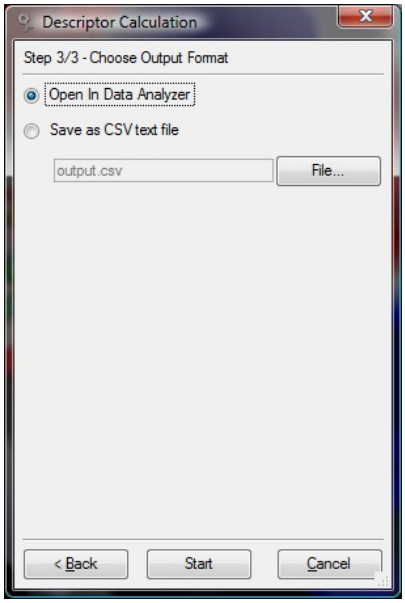


Figure 112: Choosing an output format.

The are two possibilities:

- **Open In Molegro Data Modeller**. The resulting output is directly opened as a dataset in Molegro Data Modeller for further analysis.
- **Save as CSV text file**. This will save the output as a tab-separated text file. This kind of output can be read by virtually all data processing software products.

14.4 Working with Molecular Descriptors.

There are several potential uses for molecular descriptors:

Similarity Screening

Molecular descriptors can be used to quickly screen a molecule library for compounds similar to one or more reference molecules – for instance the reference molecules could be compounds known to bind strongly to a target receptor under investigation.

It is easy to search for similar compounds in the Molegro Data Modeller using the built-in Similarity Browser (described in Section 13).

Regression Models (QSAR)

If a quantitative measure is known (for instance the experimental binding affinity), these values may be added as a column in Molegro Data Modeller. It

is then possible to create a regression model, where the molecular descriptors are used as the independent variables and the measured quantity as the target variable.

Molegro Data Modeller provides several techniques for building regression models (including Multiple Linear Regression, Partial Least Squares, Support Vector Machines, and dimensionality reduction techniques.)

Molegro Data Modeller also offers advanced methods for clustering (including k-nearest neighbours and a density based clustering scheme) and classification, and methods for detecting outliers and creating diverse subsets.

14.5 Chemical Feature Distance Matrix Descriptors

The CFDM descriptor is an unique set of descriptors created by Molegro with the following properties:

- Independence of the conformation of the molecule. They are based on the topological properties of the molecule.
- A small set of descriptors. Having a small number of descriptors makes it easier to avoid overfitting and chance correlation in the subsequent data processing.
- Based on chemical reasoning. The descriptors are based on properties which are believed to be chemically important, not on abstract graph teoretical measures.

The descriptors are calculated using the following method:

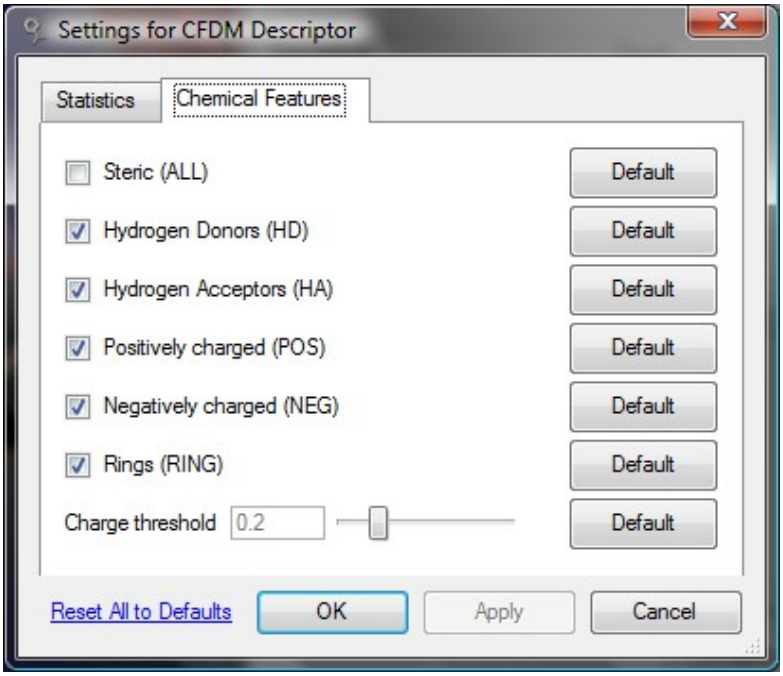


*Figure 113: The CFDM descriptors can be configured by pressing the **Configure...** button and choosing the tab **Chemical Features** in the Descriptor Calculation Wizard.*

First, all heavy atoms in the molecule are classified in one or more of the following chemical classes:

Steric – All atoms belong to this class. By default this class is not included in the CFDM calculation.

HD – All atoms with hydrogen donor capabilities.

HA – All atoms with hydrogen acceptor capabilities.

POS – All atoms with a positive charge greater than the specified threshold (default 0.2).

NEG – All atoms with a negative charge lesser than the specified threshold (default -0.2).

RING – All atoms which are part of a ring system.

Then the *topological distances* between every pair of atoms are calculated. The topological distance is the minimum number of covalent bonds connecting two atoms. This way we can extract a list of the topological distance between any two pairs of chemical classes. For instance we will have a list of distances between any atom from the **HD** class to any atom in the **HA** class.

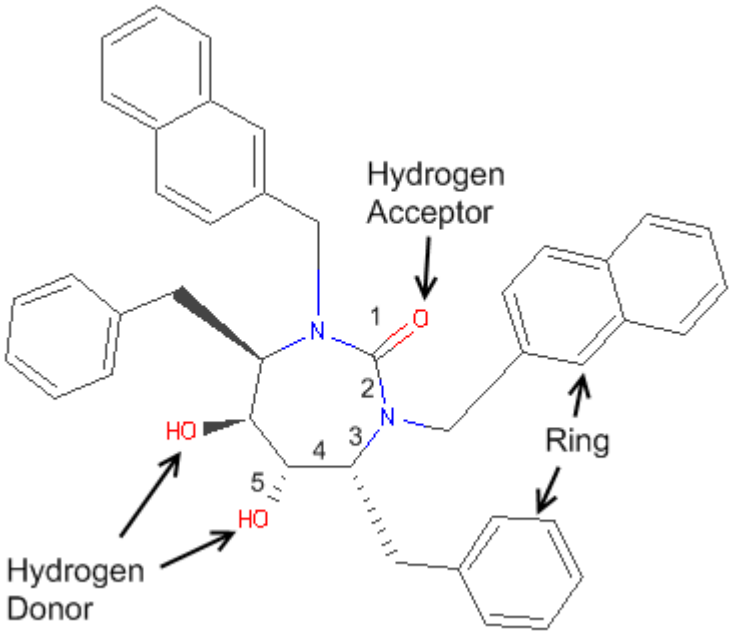


Figure 114: CFDM Calculation: the atoms in the molecule are assigned to one or more classes such as ring atom or hydrogen donor. Then all topological distances (the minimum number of covalent bonds) between any two classes are measured. For instance the minimum distance between the hydrogen acceptor atom and one of the hydrogen donor atoms is 5 as indicated on the figure. This information is summarized in a number of matrices.

This information may be summarized in a number of distance matrices. We can construct a matrix with minimum distances between any two classes, a matrix with mean distances, and a matrix with maximum distances:

	HD	HA	POS	NEG	RING
HD	3	5	0	0	1
HA		0	0	0	1
POS			0	0	0
NEG				0	0
RING					2

Table 2: Example of distance matrix.

This way we end up a total of $M \times (N \times (N+1)/2)$ numbers, where N is number of chemical classes included (per default 5: HD, HA, POS, NEG, and RING), and

M is the number of matrices (per default 3: MIN, MEAN, and MAX) giving a default of 45 descriptors. The organization of the descriptors into matrices is purely conceptual – they will be output as 45 numbers in a row vector.

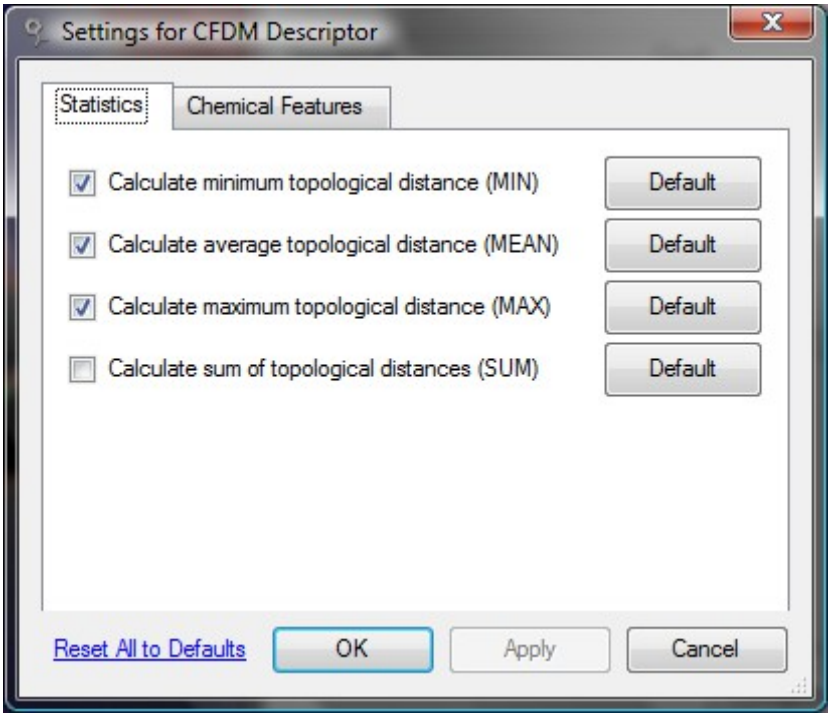


Figure 115: Choosing which matrices to include.

Finally the descriptors are named using the following convention: for instance 'HA-POS-MEAN' means the mean topological distance between any HA and POS atom.

15 Molegro Virtual Grid

Molegro Virtual Grid is a framework for distributing docking runs. It can be used for running large jobs on multiple machines in a network, or for running smaller jobs using all the available cores on a single machine. The Virtual Grid consists of two components:

- *The Controller* is a graphical application which loads a job description, and distributes the individual job units to the available resources. The controller is also able to retrieve, combine, and filter the resulting data.
- *The Agent* is a small, lightweight application that runs in the background and listens for work requests. One agent must be installed on each computer on the virtual grid. The agents receive job unit descriptions and spawn the Molegro Virtual Docker application. Notice that Molegro Virtual Docker must be installed on the agent machine together with a valid license file.

Preparing a job for distributed execution can be done automatically by MVD for certain types of jobs. A requirement is that the docking setup uses a DataSource for loading ligands (see Chapter 5). A job unit is then created for each individual ligand in the DataSource. This is a setup typical used for virtual screening. MVD cannot automatically distribute all kinds of jobs (such as docking a single ligand against multiple receptors), but it is still possible to manually create a custom grid job file that can be distributed (see Section 15.11).

Molegro Virtual Docker is a single-threaded program. This means, that when running MVD on a computer with multiple cores (as nearly all modern CPUs features) only a single core is used. However, Molegro Virtual Grid is able to run an instance of Molegro Virtual Docker for each core on the computer. Therefore it may make sense to run Molegro Virtual Grid, even if only a single computer is part of the grid. No virtual grid license is necessary to run Molegro

Virtual Grid on a single machine. Running Molegro Virtual Grid on multiple machines requires an extended license (more details about licensing can be found in Section 15.3 and Section 15.10).

We have designed Molegro Virtual Grid to be as easy as possible to install and operate. However, as with all networked software, it is important to understand a few things about network security and firewall setup.

15.1 Security Considerations

Data is transferred unencrypted between the agent and the controller.

This means that sensitive molecular data should never be transferred on the internet, since it is possible to intercept the data.

The Molegro Virtual Grid infrastructure is designed for a trusted private intranet network. If both your controllers and agents IP-numbers are in the range 10.0.0.0–10.255.255.255, 172.16.0.0–172.31.255.255, or 192.168.0.0–192.168.255.255, you are using a private network.

If you need to connect to Molegro Virtual Grid on another private network or over the internet, we strongly suggest using VPN to secure the connection (most likely this is already a requirement for accessing the private network). If you are in doubt whether your network is safe to use, please contact your network administrator. Molegro Virtual Grid use unencrypted traffic over TCP/UDP Port 45454.

15.2 Network and Firewall Issues

Molegro Virtual Grid automatically tries to detect other machines on the local network. This is done using UDP pings. If you are connecting to another network or using VPN, UDP might be blocked. In this case it is necessary to specify the IP-numbers or DNS names of the machines that make up the grid manually (see Section 15.8 for more information).

After the machines in the grid have been detected or manually specified, the actual communication between the controller and agent takes place via TCP traffic on port 45454. Notice that many modern operating systems provide some kind of firewall, which prevents software from receiving requests on arbitrary ports. The Virtual Grid Agent acts like a web server which listens for requests on port 45454. This means if a firewall is present, it must allow incoming connections for this port.

The actual details on how to configure firewall access depends on the specific operating system. For instance, on Windows Vista, the firewall can be configured using **Start Menu** | **Control Panel** | **Security** | **Allow a program though Windows Firewall** and choosing **Add port...** The following settings

can then be used: **Name**=Virtual Grid, **Port number**=45454, and **protocol**=TCP).

The physical network the machine belongs to may also have a firewall which prevents communication with other networks. If you need to communicate with a Molegro Virtual Grid on another network, we strongly suggest that you use VPN to setup the connection. Please see the previous section for more details.

15.3 Licensing

All licenses for Molegro Virtual Docker include a basic license for Molegro Virtual Grid.

The basic license makes it possible to run jobs on only one (1) agent at a time. It is possible to use any number of cores on this single machine, but only one physical machine can be used at a time. This machine can either be the same machine as Molegro Virtual Docker and the MVG controller is running on (this is useful in order to take advantage of the multiple cores), or another machine (for instance, the MVD GUI can be run on a laptop while the docking runs are executed on a more powerful desktop computer).

The extended license (which is licensed as an additional product) has no restrictions on the number of machines it controls. Together with the features in the MVG controller for combining and filtering docking results, it is possible to run very large docking runs.

15.4 Running the Agents

In order for a machine to participate in the grid and receive job units it must run the Molegro Virtual Grid Agent (and have copy of Molegro Virtual Docker together with a valid license installed). In order to start an agent on a machine, run the Virtual Grid executable:

On Windows this file is called 'virtualgrid.exe' and is located in the 'bin' directory of the Molegro Virtual Docker installation. On Linux and Mac the file is called 'virtualgrid'.

The agent writes a log file to its working directory while running. The filename for this log file is auto generated. A typical filename will be 'Log-24.11.2009-16.27.28.153.txt'. The log file is useful for detecting docking run errors and configuration errors. The log file can also be retrieved using a web-interface (see below).

15.5 The Agent GUI

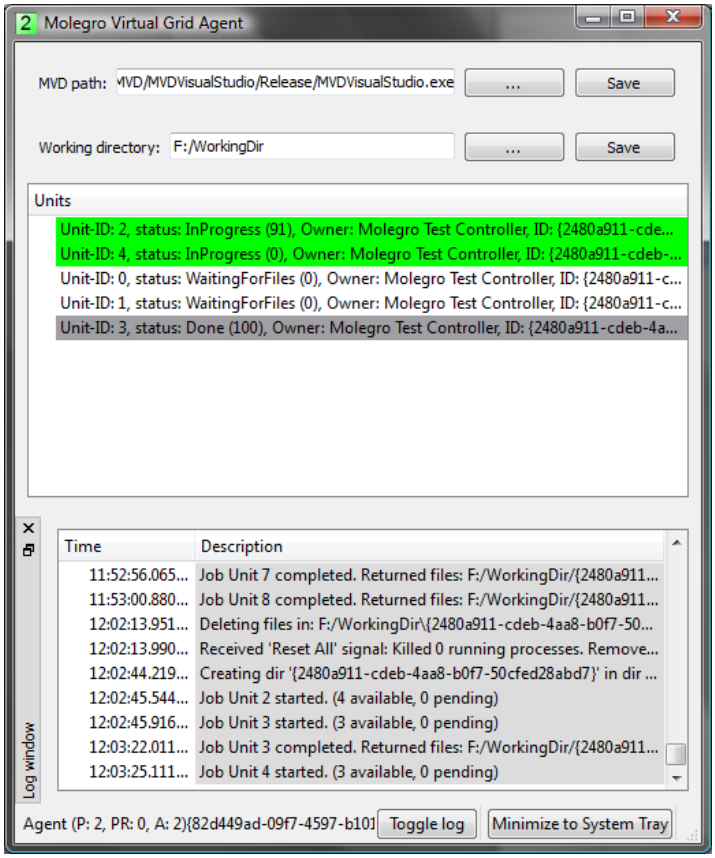

Molegro Virtual Grid Agent
MVD path: MVD/MVDVisualStudio/Release/MVDVisualStudio.exe
...
Save
Working directory: F:/WorkingDir
...
Save
Units
Unit-ID: 2, status: InProgress (91), Owner: Molegro Test Controller, ID: {2480a911-cde...
Unit-ID: 4, status: InProgress (0), Owner: Molegro Test Controller, ID: {2480a911-cdeb-...
Unit-ID: 0, status: WaitingForFiles (0), Owner: Molegro Test Controller, ID: {2480a911-c...
Unit-ID: 1, status: WaitingForFiles (0), Owner: Molegro Test Controller, ID: {2480a911-c...
Unit-ID: 3, status: Done (100), Owner: Molegro Test Controller, ID: {2480a911-cdeb-4a...
Log window
Time
Description
11:52:56.065... Job Unit 7 completed. Returned files: F:/WorkingDir/{2480a911...
11:53:00.880... Job Unit 8 completed. Returned files: F:/WorkingDir/{2480a911...
12:02:13.951... Deleting files in: F:/WorkingDir\{2480a911-cdeb-4aa8-b0f7-50...
12:02:13.990... Received 'Reset All' signal: Killed 0 running processes. Remove...
12:02:44.219... Creating dir '{2480a911-cdeb-4aa8-b0f7-50cfed28abd7}' in dir ...
12:02:45.544... Job Unit 2 started. (4 available, 0 pending)
12:02:45.916... Job Unit 3 started. (3 available, 0 pending)
12:03:22.011... Job Unit 3 completed. Returned files: F:/WorkingDir/{2480a911...
12:03:25.111... Job Unit 4 started. (3 available, 0 pending)
Agent (P: 2, PR: 0, A: 2){82d449ad-09f7-4597-b10
Toggle log
Minimize to System Tray

Figure 116: The Virtual Grid Agent GUI

In order for the agent to execute docking runs, it must know the location of the MVD executable. Either type a location in the **MVD path** settings box, or browse to the location using the **...** button. If you manually type a location, use the **Save** button to make the setting persistent.

The agent also requires some space for temporary files (files received from the controller, or docking result files). Specify a directory as above.

After the MVD path and working directory has been specified, the agent is ready to receive jobs. It can be minimized to the system tray, by pressing the button in the lower right corner. The icon (both the application icon and the system tray icon) will show the number of job units being executed. If any errors or warnings are encountered, a notification message will be shown from the tray icon. Notice that the Growl notification system (http://growl.info/) must be installed for this to work on Mac OS X.

Figure 117: Example of an Agent running in the task tray on Windows Vista. The red icon indicates that an error has occurred. The number shows the number of executing job units. The green icon is the controller icon.

The **Units** list view shows the job units currently being executed (in green), job units that are pending execution (white), and job units that are completed, but not collected from the controller yet (grey). Completed job units will be removed from the list when the controller has collected the results.

15.6 Console Mode

Per default, the agent starts up as a graphical application, but it is possible to run it as a console application as well. This is done by specifying the command-line option '-nogui'.

The following command line options are available:

-nogui	Starts the agent without a graphical user interface. This makes it possible to run the agent in the background on systems without a graphical user interface - for instance running the agent on a remote Linux system using a shell. Notice, the web interface (see Section 15.7) makes it easy to see the status and error log of the running agent.
-mvdpath	Specifies the path of the MVD executable. Example: virtualgrid -mvdpath "C:\program files\Molegro\MVD\bin\mvd.exe". The path is stored by the OS, so it only necessary to set it once. The path can also be set in the GUI.

-workingdir	Specifies the directory the grid agent use for temporary files (files received from the controller, or docking result files). Example: virtualgrid -workingdir "C:\tempdir”. The path is stored by the OS, so it only necessary to set it once. The path can also be set in the GUI.
-priority	Specifies the process priority when launching MVD instances. Notice that the process priority is normally set by the controller. Specifying an agent priority overrides the controller settings. In most cases is not necessary to set this value. The process priority specifies how the OS schedules its time when multiple processes are running simultaneously. Per default, Molegro Virtual Grid runs processes with 'below normal' priority. This means, that when running other applications on a machine with a running agent, the other applications get more CPU time, resulting in a more responsive system. It is also possible to set the priority even lower, to 'lowest (idle)', for instance to make it possible to run the agent on a desktop computer which is also used for normal work - the running jobs will only get CPU time when no processes request it. Notice that we strongly recommend against setting the priority higher than 0. The jobs will most likely not run faster, but might instead make the operating system unresponsive. The following values are available: 10 **highest** (real-time) 0 **normal** -1 **below normal** -10 **lowest** (idle) Notice that different OS's may use other process priority values internally. The values are translated by the Virtual Grid agent, so -10 is always the lowest priority no matter what OS the agent is running on.

-maxthreads	The maxthreads options determines the maximum number of simultaneous instances of MVD spawned by the agent. Under most circumstances is not necessary to set this value. This number is per default set to the number of physical CPU-cores. In some cases it might be useful to limit the number of threads - for instance in order to reserve threads for other tasks. It is not recommended to set the number higher than the number of physical CPU-cores.

15.7 Agent Web Interface

The log file for the agent can be retrieved using a web browser. Open a browser and specify the IP-number or DNS-name of the agent on port 45454: e.g. http://192.168.1.101:45454.

This will show the status and the log of the running agent. It is also possible to obtain more verbose information by appending '/debug' to the URL: e.g. http://192.168.1.101:45454/debug. Notice that the log file is also stored as a text file in the 'workingdir' directory.

The web interface can be useful for obtaining information from an agent running without a GUI, or to check if the communication between machines is being blocked by a firewall.

15.8 The Virtual Grid Controller

The controller loads a grid job description, keeps track of the available resources (the agents), and distributes the individual job units to the agents. It is also able to combine and filter the results collected from the agents.

Normally the grid controller is started directly from the docking wizard or **Tools** | **Virtual Grid Controller** menu in MVD. It is also possible to start the grid controller using the command line using the '-controller' argument, e.g. 'virtualgrid.exe -controller'. Optionally a grid job can be specified as well: e.g. 'virtualgrid.exe myjob.gridjob -controller'. If the controller is started from the docking wizard, the automatically generated grid job will be loaded on startup.

The controller must be kept running in order to distribute jobs to the agents. If the controller is closed, no further jobs are sent to the agents. It is possible to restart the job after the controller has been closed. In order to do this start the controller (e.g. from MVD using the **Tools** | **Virtual Grid Controller** or using the command line 'virtualgrid.exe -controller'). The controller is able to resume the execution of pending units and completed units are not lost.

It is also possible to set a Controller ID. This is useful if several people are running Molegro Virtual Grid controllers on the same network. The Controller

ID is a simple text label that identifies the controller user to the rest of the network. For instance, when inspecting the running job units on an agent, the controller ID is listed as the owner.

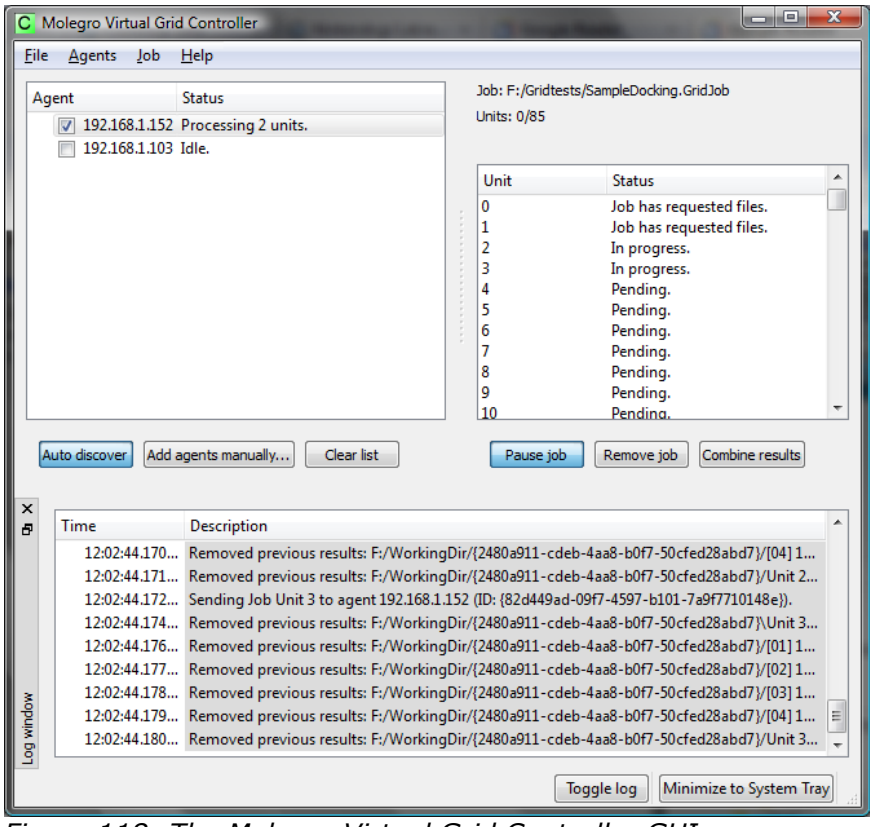


Figure 118: The Molegro Virtual Grid Controller GUI.

On Figure 118, the left panel shows a list of the agents that are currently available for processing job units. If **Auto Discover** is enabled, this list will be automatically populated with agents that can be recognized on the local network. Not all network and firewall configurations allow automatic discovery of agents: in this case it is necessary to manually add agents: this is done using the **Add agents manually...** button. It is possible to enter a list with IP-numbers or DNS-names of computers to be added, or to load a list from a text file.

When an agent appears on the list, the following actions are available using the context menu:

- **Show status**. Display statistics about the currently running jobs.
- **Reset agent**. This terminates all running jobs on the agent, and removes all temporary files produced. Notice that this will also cancel all jobs and delete all files belonging to another user. Resetting the agent can be useful in order to cancel jobs on the agent or to clean up temporary files.

- **Remove from list**. Removes the agent (useful for instance if the agent belongs to or is used by another user on the same network). If **Auto discover** is enabled, the agent might re-appear. Notice that the **Agents** menu contains an option for removing all non-responding agents.

The **Agent** menu offers a few additional options for setting the process priority: normally agents execute job units with a process priority just below the 'normal' priority. It is possible to adjust the priority to either 'normal', which is the typical priority user processes on an OS is assigned, or to 'idle' which means the job will only execute if no other process requests CPU time. Notice that the controller process priority may be overruled by the agent '-priority' command-line option.

The right panel shows the job units of the currently loaded job (only one job can be loaded at a time). Jobs created by the Docking Wizard are automatically loaded when the controller is started. It is also possible to load jobs using the **File** | **Open Job...** dialog, or by dragging a job description file onto the GUI. When **Start job** is pressed, the controller will begin dispatching units to the available agents. The **Remove job** button removes the currently loaded job from the controller. This action does not delete any files. All results are still stored on the controller computer. From the **Job** menu it is also possible to perform the **Reset job** action. Resetting a job sets all units to the 'pending' state. All produced log-files and results are deleted from disk and lost.

The following additional actions are available using either the context menu or the **Job** menu:

- **Show log file**. This will show a log file for the unit. This includes any log messages produced by running MVD on the remote agent.
- **Retry unit**. Occasionally units fail. This might be due to invalid molecule structures, MVD settings or network transmission errors. Since some types of errors (such as invalid molecule structures) can not be corrected, the controller will not automatically retry a failed unit. However, it is possible to use the 'retry unit' action to rerun the unit. Notice that the job menu also contains an option for retrying all non-completed units.

15.9 Combining Results

A distributed job will create one 'MVD results file' and a number of poses for each job unit.

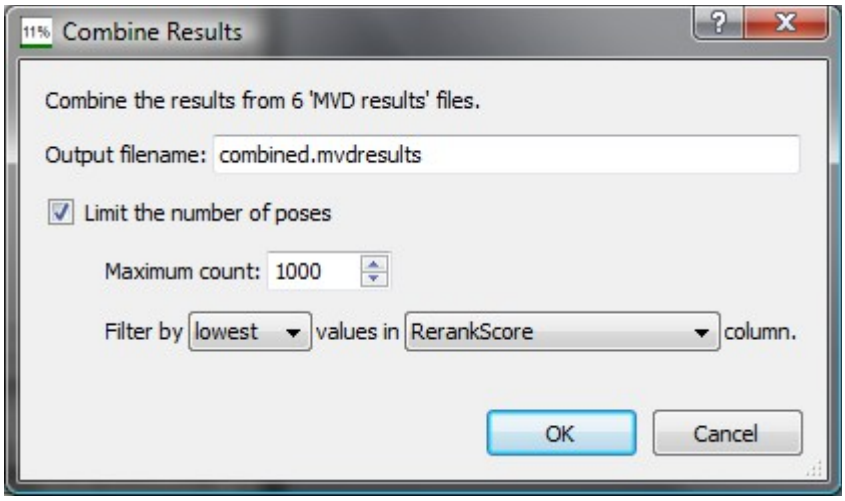


Figure 119: The interface for combining results.

Even though MVD is able to import multiple MVD results files at the same time using the Pose Organizer, larger jobs quickly become difficult to handle this way. Therefore the controller is able to combine multiple MVD results files into a single file. For larger runs (>1000 ligands) it is also possible to filter the combined results before importing them into the Pose Organizer. This is done by enabling **Limit the number of poses**, and choosing a desired number of poses. Normally the best choice would be to filter by lowest **RerankScore** or **PoseEnergy**, but it is possible to choose between all terms available in the MVD results file. Using filtering makes it possible to handle very large virtual screening runs.

When the combination and filtering has completed, a new combined MVD results file is written to the chosen location. It is possible to drag the yellow label directly onto the MVD GUI to inspect the poses with the Pose Organizer or manually import the results file in MVD.

15.10 License Management

As mentioned in the License section, the basic license makes it possible to run jobs on only one agent at a time. If an extended license has been obtained, install it by going to the **Help** menu, and choose **Install license** and specify the location of a valid Molegro Virtual Grid license file. Notice that is possible to see information about the current license by choosing **Help** | **About Molegro Virtual Grid**.

15.11 How Virtual Grid Works

The Docking Wizard in MVD makes it possible to create MVG jobs automatically when docking a DataSource with multiple ligands against a single protein

target. Other cases, such as docking a number of ligands against different receptors, require manual creation of MVG job file.

A MVG job is an XML file that describes a number of job units. The typical format is:

```
<Job id="{63647969-d2c5-496a-944a-3edcbac43d8c}" description="Job">
<Before uploadFiles="Unnamed_complex.mvdml">
  DOCKSETTINGS maxIterations=1500;runs=10;...
  EVALUATORTYPE MolDockGrid
  EVALUATOR cropdistance=0;gridresolution=0.30;...
  OPTIMIZERTYPE MSE
  OPTIMIZER populationsize=50;cavity=true;...
  LOAD "Unnamed_complex.mvdml"
</Before>
<Unit id="0" uploadFiles="ZINC02000919.mvdml">Dock
[File=ZINC02000919.mvdml]</Unit>
<Unit id="1" uploadFiles="ZINC03775575.mvdml">Dock
[File=ZINC03775575.mvdml]</Unit>
<Unit id="2" uploadFiles="ZINC00006989.mvdml">Dock
[File=ZINC00006989.mvdml]</Unit>
...
...
<After>
 DONE
</After>
</Job>
```

The **Job id** tag is a simple identifier. It can be any text string, but it must be unique. The **description** tag can be used for arbitrary remarks - it is not used by the Virtual Grid infrastructure. The grid job description consists of a <**Before**> element, any number of <**Unit**> elements, and an <**After**> element. The content of these tags are script commands for the MVD script parser. An agent will execute one unit per processing thread (per default an agent simultaneously execute one unit per physical CPU-core). Whenever an unit is executed on an agent, the script content in the <**Before**> element is executed, followed by the content in the particular <**Unit**> element being executed, finally followed by the content in the <**After**> element. The <**Before**> and <**Unit**> elements also specify an **uploadFiles** attribute, which list the files that must be uploaded to the agent before starting the job. The agent will automatically keep track of which files are produced by MVD and return them to the controller. The easiest way to create a custom grid job file is to setup a MVG job using the Docking Wizard and manually edit the saved script.

The grid job file must be saved in a file with a *.gridjob extension. All input-files must be located in a folder named 'InputFiles'. The 'InputFiles' folder must be located in the same folder as the grid job file. Notice that the 'InputFiles' cannot contain sub-folders - all molecule files must be located at the root of the 'InputFiles' folder.

When the job file is executed by the controller, two additional directories are

created. The 'JobState' directory contains one file per job unit. The extension of the file shows the current state of the job unit: e.g. "Unit1.Pending" is a job unit waiting for execution, and "Unit47.DoneAndCollected" is a job unit which has completed, and the results have been transferred from the agent back to the controller. The content of these files is the log file (including any log messages produced by MVD during the docking run).

The 'OutputFiles' directory contains the files that have been transferred from the different agents and back to the controller. This includes the molecular structure files (the poses) and the MVD docking results.

16 Help

16.1 PDF Help

The documentation for Molegro Virtual Docker is available as a PDF file. In order to invoke the PDF help using the built-in PDF reader, choose **Help** | **Molegro Virtual Docker Manual** from the menu bar. The executable for the PDF reader can be specified in the Preferences.

16.2 Tip of the Day

A 'Tip-of-the-Day' dialog (see Figure 120) providing useful tips on how to use Molegro Virtual Docker is available.

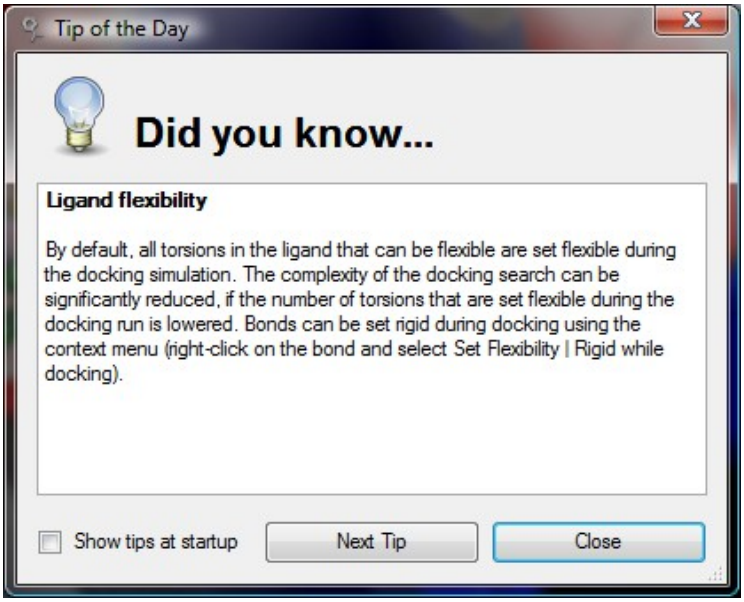

Tip of the Day
Did you know...
Ligand flexibility
By default, all torsions in the ligand that can be flexible are set flexible during the docking simulation. The complexity of the docking search can be significantly reduced, if the number of torsions that are set flexible during the docking run is lowered. Bonds can be set rigid during docking using the context menu (right-click on the bond and select Set Flexibility | Rigid while docking).
Show tips at startup
Next Tip
Close

Figure 120: 'Tip of the Day' dialog.

The dialog can be manually invoked from the **Help** menu or automatically shown on startup. The automatic startup setting can be toggled in the dialog or from the general **Preferences** dialog.

16.3 The Molegro Website

The Molegro website also offers certain help facilities. Please visit www.molegro.com to see our FAQs and other information available.

16.4 Technical Support

Technical support is available for commercial licenses (industrial and academic) only. To obtain additional support, send an email to support@clcbio.com.

17 Script Interface

17.1 Using the Script Interface

The default behavior for docking molecules in Molegro Virtual Docker, is to start the application, load and prepare the molecules, and invoke the **Docking Wizard**.

The **Docking Wizard** guides the user through the different settings for the simulation, and then creates a small script file which contains instructions on how the docking should proceed.

The default behavior for the **Docking Wizard** is then to spawn a script interpreter (in another process – the script interpreter and the main application runs completely separated) and execute the script.

However greater flexibility is possible by writing custom scripts: for instance this makes it possible to dock a number of ligands against several distinct targets. It is also possible to split large docking runs into several scripts and run them on different machines.

Notice: A MVD script job basically runs in a single thread. This means that as such, MVD will not utilize multiple CPU's (or dual-core processors). However by splitting the job into two (or more) jobs and running them concurrently all available CPU's can be utilized.

17.2 Running a Text-file Script

Text-file scripts are ordinary text files saved with the `mvdscript` file extension.

In order to run a text-file script, simply start MVD with the text-file script name as the argument:

Example: `mvd docktest.mvdscript`

This will spawn the Script Progress GUI with information on how the script parsing is progressing:

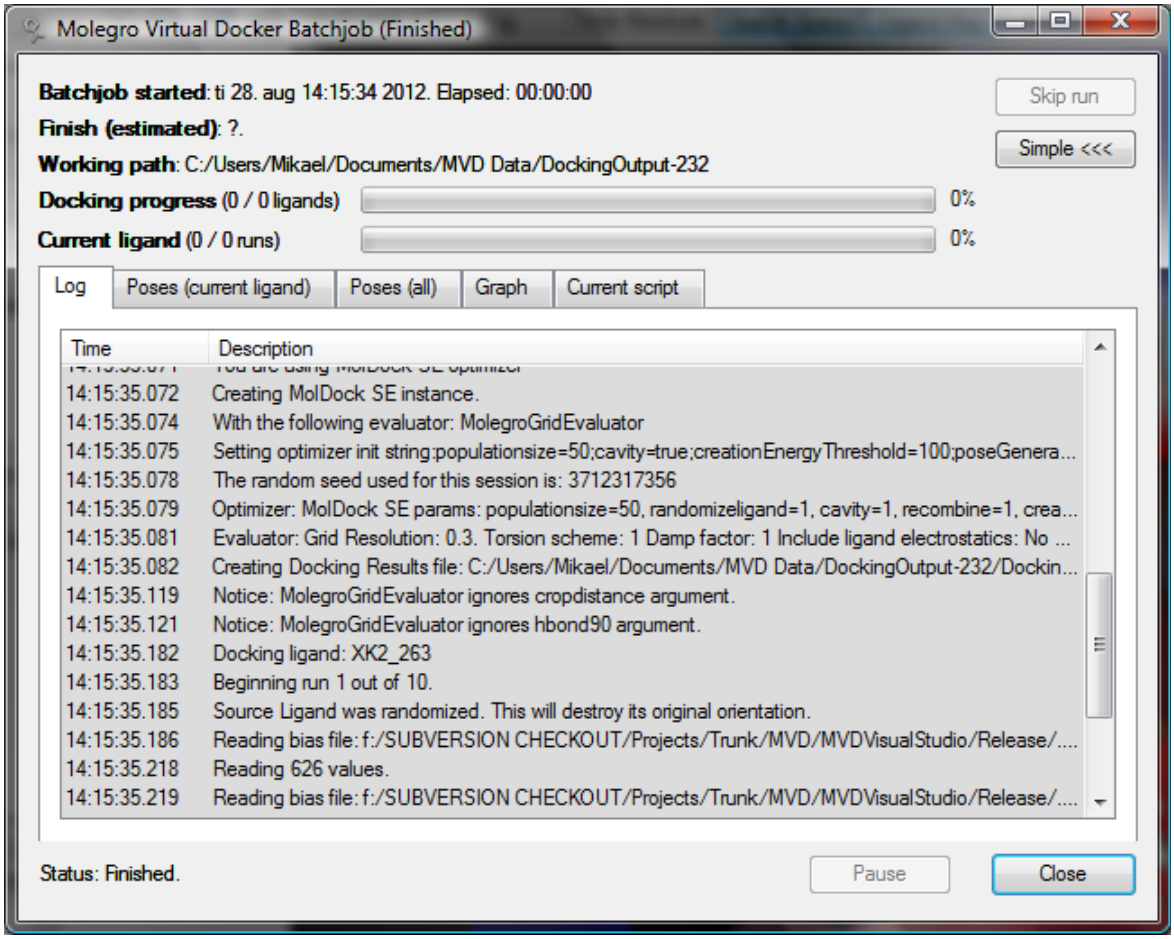

Molegro Virtual Docker Batchjob (Finished)
Batchjob started: ti 28. aug 14:15:34 2012. Elapsed: 00:00:00
Finish (estimated): ?.
Working path: C:/Users/Mikael/Documents/MVD Data/DockingOutput-232
Docking progress (0 / 0 ligands)
Current ligand (0 / 0 runs)
Skip run
Simple <<<
0%
0%
Log
Poses (current ligand)
Poses (all)
Graph
Current script
Time
Description
14:15:35.072 Creating MolDock SE instance.
14:15:35.074 With the following evaluator: MolegroGridEvaluator
14:15:35.075 Setting optimizer init string:populationsize=50;cavity=true;creationEnergyThreshold=100;poseGenera...
14:15:35.078 The random seed used for this session is: 3712317356
14:15:35.079 Optimizer: MolDock SE params: populationsize=50, randomizeligand=1, cavity=1, recombine=1, crea...
14:15:35.081 Evaluator: Grid Resolution: 0.3. Torsion scheme: 1 Damp factor: 1 Include ligand electrostatics: No ...
14:15:35.082 Creating Docking Results file: C:/Users/Mikael/Documents/MVD Data/DockingOutput-232/Dockin...
14:15:35.119 Notice: MolegroGridEvaluator ignores cropdistance argument.
14:15:35.121 Notice: MolegroGridEvaluator ignores hbond90 argument.
14:15:35.182 Docking ligand: XK2_263
14:15:35.183 Beginning run 1 out of 10.
14:15:35.185 Source Ligand was randomized. This will destroy its original orientation.
14:15:35.186 Reading bias file: f:/SUBVERSION CHECKOUT/Projects/Trunk/MVD/MVDVisualStudio/Release/....
14:15:35.218 Reading 626 values.
14:15:35.219 Reading bias file: f:/SUBVERSION CHECKOUT/Projects/Trunk/MVD/MVDVisualStudio/Release/....
Status: Finished.
Pause
Close

Figure 121: Script Progress GUI.

Notice: MVD scripts need to have the `.mvdscript` file extension. Otherwise, the script file will not be recognized (and parsed) by MVD.

It is also possible to start a script job with no graphical user interface (without the script parsing progress dialog). This can done by using the `-nogui` command line argument:

Example: `mvd docktest.mvdscript -nogui`

Notice: If you intend to run background jobs on remote Linux/X11 systems, use the `-nogui` argument. Otherwise the system might kill the process when the user logs off (because the X11 server might be shutdown).

17.3 Examples of Common Script Jobs

This section contains some examples of common script jobs. Another useful way of exploring the MVD script syntax is to inspect the script files generated

by the **Docking Wizard**: these files are stored as ordinary MVD script files in the specified directory and can be opened using a standard text editor.

A complete list of commands can be found in Appendix XI: Script Commands.

Docking a Single Complex

```
// Init
DOCKSETTINGS maxIterations=1000;runs=1;MaxPoses=5
EVALUATOR cropdistance=0;hbond90=true
OPTIMIZER cavity=true;popsize=50;crossoverrate=0.9;keepmaxposes=5

// Dock
LOAD 3PTB.MVDML
RMSD ligand[0]
DOCK
```

Docking Multiple Complexes

```
// Init
DOCKSETTINGS maxIterations=1000;runs=1;MaxPoses=5
EVALUATOR cropdistance=0;hbond90=true
OPTIMIZER cavity=true;popsize=50;crossoverrate=0.9;keepmaxposes=5

// Dock
FOR $MVDML IN 3PTB,1HVR,1LIC,1TMN
   // $MVDML will be replaced by the appropriate value in the loop
   LOAD C:\BENCHMARK\$MVDML.mvdml
   RMSD ligand[0]
   DOCK
   NEW
ENDFOR
```

Splitting Docking Runs Into Multiple Runs

This script can be used to divide the workload between different machines.

```
// Init with appropriate settings first...
DOCKSETTINGS maxIterations=1000;runs=10;MaxPoses=5
EVALUATOR cropdistance=0;hbond90=true
OPTIMIZER cavity=true;popsize=50;crossoverrate=0.9;keepmaxposes=5
// For machine 1:
LOAD C:\BENCHMARK\1HVR.mvdml
IMPORT LIGAND[0-99] FROM DB.sdf
DOCK

// For machine 2:
LOAD C:\BENCHMARK\1HVR.mvdml
IMPORT LIGAND[100-199] FROM DB.sdf
DOCK
```

17.4 Running the Script Interface Interactively

MVD can also run in interactive mode.

In this mode the MVD application starts and waits for user input from the command line (i.e. it reads and writes from the standard input and output which can be piped).

To start MVD in interactive mode, use the following syntax:

Example: `mvd -interactive`

The purpose of the interactive mode is to allow scripting languages capable of writing to and from the standard input and output of a program to control the docking process. This can be useful for automating larger docking runs.

When in interactive mode, MVD will send an '[DONE]: <command>' after each command has been interpreted.

17.5 Running the Script Interface From Python

A small Python wrapper is provided in:

`MVD/Scripting/Python/MvdWrapper.py`

The wrapper encapsulates the various script commands in a small object, `MVDWrapper`.

The wrapper spawns a new MVD process when the object is instantiated and runs MVD in interactive mode to pass commands to it.

The process can be terminated by calling `exit` on it.

In order to use the wrapper, copy the `MvdWrapper.py` file to the same location as your Python-script (or install it in a globally accessible location) and `import` it at the top of you script.

Notice: The Python wrapper requires Python 2.4 (or above).

The following example is taken from:

`MVD/Scripting/PythonWrapper/SimpleDockingTest.py`

```
import os
import MvdWrapper

# create an output dir
outputPath = 'outputData'
complex = '1hvr'

if (not os.path.exists( outputPath )):
     os.mkdir( outputPath )

if os.path.exists( outputPath ) and os.path.isdir( outputPath ):
     print 'Created outputPath: ' + outputPath
else:
     raise IOError, 'could not create path' + outputPath
```

```
# Now start the wrapper...
# Remember to change the path to the executable in the line below:
mvd = MvdWrapper.MvdWrapper("C:/Program Files/Molegro/MVD/bin/mvdconsole.exe",
gui=True)
mvd.info("testing")
mvd.random(123232)                        # set the seed
mvd.cd(outputPath)                        # change to output path
mvd.download(complex, complex + ".pdb")   # download from pdb.org
mvd.importFrom("All", complex + ".pdb")   # import into workspace
mvd.rmsd("ligand[0]")                     # set a ligand as a rmsd reference
mvd.dock("")                              # start the docking
mvd.exit()
```

Notice for Windows Users:

In order to use the Python wrapper you must install the "Python for Windows extensions" which can be downloaded from:

`http://sourceforge.net/project/showfiles.php?group_id=78018`

Notice that you must download the version which targets your specific Python version. Also notice that in order to communicate through pipes with the MVD application be sure to instantiate with a reference to the 'MVDConsole.exe' instead of the standard 'MVD.exe' application. Use:

```
mvd = MvdWrapper.MvdWrapper("C:\Program Files\Molegro\MVD\Bin\MVDConsole.exe")
```

instead of: `mvd = MvdWrapper.MvdWrapper("C:\Program Files\Molegro\MVD\Bin\MVD.exe")`

18 Appendix I: MolDock Scoring Function

The MolDock scoring function (MolDock Score) used by MVD is derived from the PLP scoring functions originally proposed by Gehlhaar et al. [GEHLHAAR 1995,1998] and later extended by Yang et al. [YANG 2004]. The MolDock scoring function further improves these scoring functions with a new hydrogen bonding term and new charge schemes. The docking scoring function, E_{score}, is defined by the following energy terms:

$$E_{score} \quad E_{inter} \quad E_{intra}$$

where E_{inter} is the ligand-protein interaction energy:

$$E_{inter} \underset{i \ ligand \ j \ protein}{} E_{PLP}(r_{ij}) \quad 332.0 \frac{q_i q_j}{4 r_{ij}^{\ 2}}$$

The summation runs over all heavy atoms in the ligand and all heavy atoms in the protein including any cofactor atoms and water molecule atoms that might be present. The E_{PLP} term is a piecewise linear potential described below. The second term describes the electrostatic interactions between charged atoms. It is a Coulomb potential with a distance-dependent dielectric constant given by: $D(r) = 4r$. The numerical value of 332.0 fixes the units of the electrostatic energy to kilocalories per mole. To ensure that no energy contribution can be higher than the clash penalty the electrostatic energy is cut-off at the level corresponding to a distance of 2.0 Å for distances less than 2.0 Å. Notice that although the electrostatic energy contribution has the theoretically predicted

magnitude, the other energy terms are empirically motivated and the total energy does not necessarily correlate with the true binding affinity. The charges are set according to the scheme listed in Table 3. Metal ions are assigned a charge of +1 (e.g. *Na*) or +2 (e.g. *Zn*, *Ca*, *Fe*).

charge	ligand atoms	protein atoms
0.5	N atoms in $-C(NH_2)_2$	His (ND1/NE2)
		Arg (NH1/NH2)
1.0	N atoms in $-N(CH_3)_2$, $-(NH_3)$	Lys (N)
-0.5	O atoms in $-COO$, $-SO_4$,	Asp (OD1/OD2)
	$-PO_2$, $-PO_2-$	Glu (OE1/OE2)
-0.66	O atoms in $-PO_3$	
-0.33	O atoms in $-SO_3$	
-1.0	N atoms in $-SO_2NH$	

Table 3: Charge templates.

E_{PLP} is a "piecewise linear potential" using two different sets of parameters: One set for approximating the steric (Van der Waals) term between atoms, and another stronger potential for hydrogen bonds. The linear potential is defined by the following functional form:

$$E_{PLP}(0) = A_0,\ E_{PLP}(R_1) = 0,\ E_{PLP}(R_2) = E_{PLP}(R_3) = A_1,\ E_{PLP}(r) = 0 \text{ for } r \geq R_4$$

and is linearly interpolated between these values. The parameters used here (see Table 4) were adopted from GEMDOCK [YANG 2004].

	A_0	A_1	R_1	R_2	R_3	R_4
hydrogen bond	20.0	-2.5	2.3	2.6	3.1	3.6
steric	20.0	-0.4	3.3	3.6	4.5	6.0

Table 4: PLP parameters.

A bond is considered a hydrogen bond if one of the atoms can *donate* a hydrogen atom and the other atom can *accept* it. The atom types are assigned according to the scheme shown in Table 5.

type	atoms
acceptor	N and O (with no Hs attached)
donor	N and S (with one or more Hs attached)
both	O (with one H attached) or O in water molecules
nonpolar	all other atoms

Table 5: Hydrogen bond types.

The PLP hydrogen bond term mentioned above only depends on the distance between atoms. In order to take into account the directionality of the hydrogen bonding, the geometry of the hydrogen bond is examined and the following factor H_{factor} is multiplied to the PLP hydrogen bond strength:

$$H_{factor} = \Phi(\quad_{D\text{-}H\text{-}A};90°;150°)\cdot\Phi\ (\quad_{H\text{-}A\text{-}AA};90°;100°)\cdot\ \Phi\ (\quad_{D\text{-}A\text{-}AA};90°;100°)$$

Here *AA* (Acceptor Antecedent) denotes a heavy atom connected to the acceptor (A), *D* denotes the donor and *H* is the donated hydrogen atom. The ramp function Φ is defined as $\Phi(A;A_{min};A_{max}) = 0$ for $A \leq A_{min}$ and $\Phi(A;A_{min};A_{max}) = 1$ for $A \geq A_{max}$ and is linearly interpolated between these values for $A_{min} < A < A_{max}$. If it is not possible to calculate one of these factors it is omitted. This is for example the case for hydroxyl rotors where the exact location of the hydrogen is not investigated during docking, and the two first factors cannot be calculated. The angle checks above were motivated by the approach taken by McDonald and Thornton [MCDONALD 1994].

E_{intra} is the internal energy of the ligand:

$$E_{intra} \qquad\qquad E_{PLP}(r_{ij}) \qquad\qquad A[1 \quad \cos(m \qquad\quad_0)] \qquad E_{clash}$$
$$i\ \ ligand\ \ j\ \ ligand \qquad\qquad flexible\ bonds$$

The double summation is between all atom pairs in the ligand excluding atom pairs which are connected by two bonds or less. The second term is a torsional energy term, parameterized according to the hybridization types of the bonded atoms (see Table 6). θ is the torsional angle of the bond. Notice that this angle is not necessarily uniquely determined. The average of the torsional energy bond contribution was used if several torsions could be determined. The last

term, E_{clash}, assigns a penalty of 1000 if the distance between two heavy atoms (more than two bonds apart) is less than 2.0 Å. Thus, E_{clash} term punishes infeasible ligand conformations. Finally, if a ligand heavy atom is located outside the binding site region (defined by the search space sphere) a constant penalty of 10000 is assigned to the total energy (notice: this penalty scheme is only used for the grid-based version of the MolDock Score).

	θ_0	m	A
sp^2-sp^3	0.0	6	1.5
sp^3-sp^3	∏	3	3.0
*sp^2-sp^2	0	2	3.0

Table 6: Torsional parameters.

(the sp^2-sp^2 term is not enabled by default)*

Terms in the '.mvdresults' file

After MVD has predicted one or more promising poses using the MolDock score, it calculates several additional energy terms. All of these terms are stored in the 'DockingResults.mvdresults' file at the end of the docking run.

The 'rerank score' is a linear combination of these terms, weighted by the coefficients given in the 'RerankingCoefficients.txt'.

A '.mvdresults' file is not meant to be interpreted or inspected manually. Instead it should be opened in MVD (either by dragging it onto the workspace or by selecting 'File | Import Docking Results (*.mvdresults)...'. It is also possible to open the file in Molegro Data Modeller in order to create new regression models based on the energy terms in the file.

The following table explains the different terms in a '.mvdresults' file:

Textual Information	
Ligand	The name of the ligand the pose was created from.
Name	The internal name of the pose (a concatenation of the pose id and ligand name).
Filename	The file containing the pose.
Workspace	The workspace (.mvdml-file) containing the protein. (Notice: This entry appears in the header of the mvdresults file)
Run	When running multiple docking runs for each ligand, this field contains the docking run number.
Energy terms	

(total)	
Energy	The MolDock score (arbitrary units). Notice that this value is always calculated using the non-optimized MolDock score (and hence may differ from the PoseEnergy below which may use interpolation on precalculated grids).
RerankScore	The reranking score (arbitrary units).
PoseEnergy	The score actually assigned to the pose during the docking. Notice that since the score is calculated by the scoring function chosen in the Docking Wizard, there may be small differences to the MolDock score reported in the 'Energy' entry (for instance when using the grid-based version of the MolDock score the grid interpolation may result in slighty different energies as compared to the non-grid MolDock score version)
SimilarityScore	Similarity Score (if docking templates are enabled).
LE1	Ligand Efficiency 1: MolDock Score divided by Heavy Atoms count.
LE3	Ligand Efficiency 3: Rerank Score divided by Heavy Atoms count.
Energy terms (contributions)	
E-Total	The total MolDock Score energy is the sum of internal ligand energies, protein interaction energies and soft penalties.
E-Inter total	The total MolDock Score interaction energy between the pose and the target molecule(s).
E-Inter (cofactor - ligand)	The total MolDock Score interaction energy between the pose and the cofactors. (The sum of the steric interaction energies calculated by PLP, and the electric and hydrogen bonding terms below)
Cofactor (VdW)	The steric interaction energy between the pose and the cofactors calculated using a LJ12-6 approximation. *Notice: This term is not used by the MolDock score*
Cofactor (elec)	The electrostatic interaction energy between the pose and the cofactors.
Cofactor (hbond)	The hydrogen bonding interaction energy between the pose and the cofactors (calculated by PLP).
E-Inter (protein - ligand)	The MolDock Score interaction energy between the pose and the protein. (Equal to Steric+HBond+Electro+ElectroLong below)
Steric	Steric interaction energy between the protein and the ligand (calculated by PLP).
HBond	Hydrogen bonding energy between protein and ligand (calculated by PLP).
Electro	The short-range (r<4.5Å) electrostatic protein-ligand interaction energy.
ElectroLong	The long-range (r>4.5Å) electrostatic protein-ligand interaction energy.
NoHBond90	This is the hydrogen bonding energy (protein-ligand) as calculated if the directionality of the hbond was not taken into account. *Notice: This term is not used by the MolDock score*
VdW (LJ12-6)	Protein steric interaction energy from a LJ 12-6 VdW potential approximation. *Notice: This term is not used by the MolDock score*

E-Inter (water - ligand)	The MolDockScore interaction energy between the pose and the water molecules.
E-Intra (tors, ligand atoms)	The total internal MolDockScore energy of the pose.
E-Intra (steric)	Steric self-interaction energy for the pose (calculated by PLP).
E-Intra (hbond)	Hydrogen bonding self-interaction energy for the pose (calculated by PLP). *Notice: This is a non-standard term and is zero by default – it must be enabled by specifying the ''internalhbond=true' option to the EVALUATOR initializer list in a MVDScript file or by enabling the 'Internal HBond' option in the Docking Wizard.*
E-Intra (elec)	Electrostatic self-interaction energy for the pose. *Notice: This is a non-standard term and is zero by default – it must be enabled by specifying the 'ligandes=true' option to the EVALUATOR initializer list in a MVDScript file or by enabling the 'Internal ES' option in the Docking Wizard.*
E-Intra (tors)	Torsional energy for the pose.
E-Intra (sp2-sp2)	Additional sp2-sp2 torsional term for the pose . *Notice: This is a non-standard term and is zero by default – it must be enabled by specifying the 'sp2sp2bond=true' option to the EVALUATOR initializer list in a MVDScript file or by enabling the 'Sp2-Sp2 Torsions' option in the Docking Wizard. Also notice that only bonds that are chosen rotatable are taken into account when calculating the torsional terms for the ligand – and sp2-sp2 bonds are most often double bonds which per default are held fixed in the docking simulation.*
E-Intra (vdw)	Steric self-interaction energy for the pose (calculated by a LJ12-6 VdW approximation). *Notice: This term is not used by the MolDock score*
E-Solvation	The energy calculated from the implicit solvation model. *Notice: This energy term is considered to be an* experimental *feature only. Per default it is NOT calculated. In order to try this feature, the protein must be prepared by calling the 'prep solvation' command from the console. As of now, we recommend not to use it.*
E-Soft Constraint Penalty	The energy contributions from soft constraints.
Static terms	
Torsions	The number of (chosen) rotatable bonds in the ligand.
HeavyAtoms	Number of heavy atoms.
MW	Molecular weight (in dalton).
C0	Obsolete constant term. This value is always 1. (Older versions of the Data Analyser required an explicit constant column, in order to include a constant term in the fit – it is only included for backward compatibility)
CO2minus	Number of Carboxyl groups in ligand.
Csp2	Number of Sp2 hybridized carbon atoms in ligand.
Csp3	Number of Sp3 hybridized carbon atoms in ligand.
DOF	Degrees of internal rotational freedom. As of now this is the number of chosen rotatable bonds in the ligand and is thus equal to the 'Torsions' term. It is supposed to reflect how many rotational degrees of freedom are lost upon binding. Future work may include a more advanced model where the actual conformation is

	inspected in order to determine whether rotational degrees of freedom are lost.
N	Number of nitrogen atoms in ligand.
Nplus	Number of positively charged nitrogen atoms in ligand.
OH	Number of hydroxyl groups in ligand.
OPO32minus	Number of PO_4^{2-} groups in ligand.
OS	Number of ethers and thioethers in ligand.
carbonyl	Number of Carbonyl groups in ligand.
halogen	Number of Halogen groups in ligand.
Other terms	
RMSD	The RMS deviation from a reference ligand (if available).

19 Appendix II:PLANTS Scoring Function

The PLANTS scoring function (PLANTS Score) used by MVD is derived from the PLANTS scoring function originally proposed by Korb et al. [KORB 2009].

The MolDock scoring function further improves these scoring functions with a new hydrogen bonding term and new charge schemes.

The docking scoring function, $E_{plantsscore}$, is defined by the following energy terms:

$$E_{plantsscore} = f_{PLP} + f_{clash} + f_{tors} + c_{site} - 20$$

where f_{PLP} is a piecewise linear potential taking into account protein-ligand interactions. The PLP potential is similar to the one used by MolDock Score but here more interaction types (repulsive, buried, nonpolar, hydrogen bonding and metal) are taken into account whereas MolDock Score only has two – one for steric interactions and one for hydrogen bonding interactions. The PLP interaction parameters used by MVD are: $w_{plp\text{-}hb}$ = -2, $w_{plp\text{-}met}$ = -4, $w_{plp\text{-}bur}$ = -0.05, $w_{plp\text{-}nonp}$ = -0.4, $w_{plp\text{-}rep}$ = 0.5, w_{tors} = 1 (see [KORB 2009] for details).

The ligand clash and torsional potentials, f_{clash} and f_{tors} take into account internal ligand clashes and torsional contributions for the flexible bonds in the ligand (see [KORB 2009] for specific implementation details).

The c_{site} term specifies a penalty that is calculated if a ligand conformation (pose) is located outside the binding site (defined by the search space sphere). For each heavy atom located outside the binding site, a constant value of 50 is added to the c_{site} term. In addition, a quadratic penalty is added if the ligands reference point (i.e. the origin of the ligand's coordinate system) is located outside the search space sphere [KORB 2009].

The -20 energy offset was originally needed for the PLANTS search algorithm

and is included here in order for PLANTS scores to be comparable with the original PLANTS implementation.

Implementation Details

The implementation of the PLANTS scoring function in MVD differs from the original PLANTS implementation in the following two cases:

1) The original PLANTS implementation ignores default parameters for the Tripos torsional potential when handling 'dummy' or 'S.o2' typed atoms. This means that contributions for these atom types are not taken into account in the torsional potential. By default, the MVD implementation takes all atom types into account (non matching types will use default settings as described by Clark et al. [CLARK 1989].
2) The penalty term, c_{site}, used by PLANTS is not well-suited for the MolDock Optimizer or the MolDock SE search algorithms. By default, this penalty term is replaced by penalty scheme where a constant penalty of 10000 is assigned to the total energy if a ligand heavy atom is located outside the binding site region (defined by the search space sphere).

The settings for original PLANTS implementation can be used in MVD by adding the 'originalplants=true' parameter option to the EVALUATOR script command (see Appendix XI: Script Commands for more details).

Usage From GUI

In order to use the PLANTS scoring function choose '**Scoring function** -> **Score** -> **PLANTS Score**' from the **Docking Wizard**.

The following parameter can be set:

Include hydrogens in torsion term toggles whether or not hydrogens should be included when calculating the Tripos torsion potential, f_{tors}

Usage When Scripting

To use the PLANTS scoring function, the EVALUATORTYPE script command has to be set. Moreover, specific scoring function parameters are set by the EVALUATOR script command (see Appendix XI: Script Commands for more details).

20 Appendix III: MolDock Optimizer

The MolDock Optimizer search algorithm (MolDock Optimizer) used in MVD is based on an *evolutionary algorithm* [MICHALEWICZ 1992,2000].

Evolutionary algorithms (EAs) are iterative optimization techniques inspired by Darwinian evolution theory. In EAs, the evolutionary process is simplified and thus it has very little in common with real world evolution. Nevertheless, during the last fifty years EAs have proved their worth as powerful optimization techniques that can assist or replace traditional techniques when these fail or are inadequate for the task to be solved.

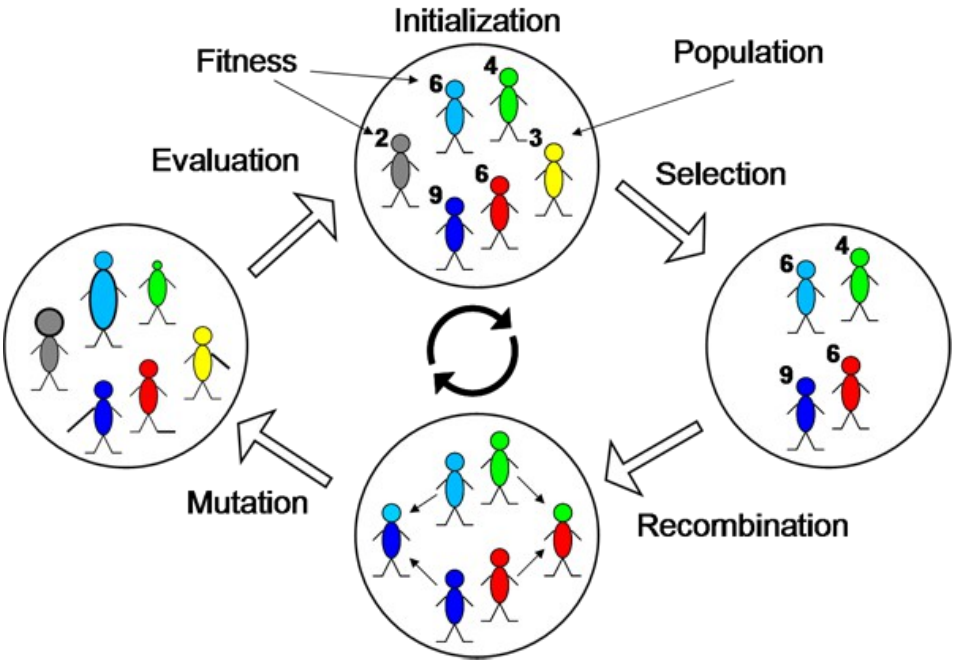


Figure 122: Outline of evolutionary algorithm.

Basically, an EA consists of a population of individuals (candidate solutions), which is exposed to random variation by means of variation operators, such as mutation and recombination. The individual being altered is often referred to as the *parent* and the resulting solution after modification is called the *offspring*. Sometimes more than one parent is used to create the offspring by recombination of solutions, which is also referred to as *crossover*. Figure 122 below shows an outline of the *evolutionary process* taking place in EAs.

Guided Differential Evolution

The guided differential evolution algorithm (MolDock Optimizer) used in MVD is based on an EA variant called differential evolution (DE). The DE algorithm was introduced by Storn and Price in 1995 [STORN 1995]. Compared to more widely known EA-based techniques (e.g. genetic algorithms, evolutionary programming, and evolution strategies), DE uses a different approach to select and modify candidate solutions (individuals). The main innovative idea in DE is to create offspring from a weighted difference of parent solutions.

The DE works as follows: First, all individuals are initialized and evaluated according to the docking scoring function (fitness function) used. Afterwards, the following process will be executed as long as the termination condition is not fulfilled: For each individual in the population, an offspring is created by adding a weighted difference of the parent solutions, which are randomly selected from the population. Afterwards, the offspring replaces the parent, if and only if it is more fit. Otherwise, the parent survives and is passed on to the next generation (iteration of the algorithm).

Additionally, guided differential evolution may use a cavity prediction algorithm (introduced in Appendix IV: Cavity Prediction) to constrain predicted conformations (poses) during the search process. More specifically, if a candidate solution is positioned outside the cavity, it is translated so that a randomly chosen ligand atom will be located within the region spanned by the cavity. Naturally, this strategy is only applied if a cavity has been found. If no cavities are reported, the search procedure does not constrain the candidate solutions.

One of the reasons why DE works so well is that the variation operator exploits the population diversity in the following manner: Initially, when the candidate solutions in the population are randomly generated the diversity is large. Thus, when offspring are created the differences between parental solutions are big, resulting in large step sizes being used. As the algorithm converges to better solutions, the population diversity is lowered, and the step sizes used to create offspring are lowered correspondingly. Therefore, by using the differences between other individuals in the population, DE automatically adapts the step sizes used to create offspring as the search process converges toward good solutions.

Representation

Only ligand properties are represented in the individuals since the protein remains rigid during the docking simulation. Thus, a candidate solution is encoded by an array of real-valued numbers representing ligand position, orientation, and conformation as Cartesian coordinates for the ligand translation, four variables specifying the ligand orientation (encoded as a rotation vector and a rotation angle), and one angle for each flexible torsion angle in the ligand (if any).

Initialization

Each individual in the initial population is assigned a random position within the search space region (defined by the user).

Initializing the orientation is more complicated: By just choosing uniform random numbers for the orientation axis (between -1.0 and 1.0 followed by normalization of the values to form a unit vector) and the angle of rotation (between -180° and +180°), the initial population would be biased towards the identity orientation (i.e. no rotation). To avoid this bias, the algorithm by Shoemake et al. [SHOEMAKE 1992] for generating uniform random quaternions is used and the random quaternions are then converted to their rotation axis/rotation angle representation.

The flexible torsion angles (if any) are assigned a random angle between -180° and +180°.

Algorithmic Settings

In MVD, the following default parameters are used for the guided differential evolution algorithm: *population size = 50*, *crossover rate = 0.9*, and *scaling factor = 0.5*. These settings have been found by trial and error, and are generally found to give the best results across a test set of 77 complexes.

21 Appendix IV: Cavity Prediction

In order to determine the potential binding sites, a grid-based cavity prediction algorithm has been developed. The cavity prediction algorithm works as follows:

First, a discrete grid with a resolution of 0.8 Å covering the protein is created. At every grid point a sphere of radius 1.4 Å is placed. It is checked whether this sphere will overlap with any of the spheres determined by the Van der Waals radii of the protein atoms. Grid points where the probe clashes with the protein atom spheres will be referred to as part of the inaccessible volume, all other points are referred to as accessible.

Second, each accessible grid point is checked for whether it is part of a cavity or not using the following procedure: From the current grid point a random direction is chosen, and this direction (and the opposite direction) is followed until the grid boundaries are hit, checking if an inaccessible grid point is hit on the way. This is repeated a number of times, and if the percentage of lines hitting an inaccessible volume is larger than a given threshold, the point is marked as being part of a cavity. By default 16 different directions are tested, and a grid point is assumed part of a cavity if 12 or more of these lines hit an inaccessible volume. The threshold can be tuned according to how enclosed the found cavities should be. A value of 0% would only be possible far from the protein as opposed to a value of 100% corresponding to a binding site buried deeply in the protein.

The final step is to determine the connected regions. Two grid points are connected if they are neighbours. Regions with a volume below 10.0 $Å^3$ are discarded as irrelevant (the volume of a connected set of grid points is estimated as the number of grid point times the volume of a unit grid cell). The cavities found are then ranked according to their volume.

22 Appendix V: Clustering Algorithm

The multiple poses returned from a docking run are identified using the following procedure:

- During the docking run, new candidate solutions (poses) scoring better than parental solutions (see Appendix III: MolDock Optimizer for details) are added to a temporary pool of docking solutions.
- If the number of poses in the pool is higher than 300, a clustering algorithm is used to cluster all the solutions in the pool (see below). The clustering is performed on-line during the docking search and when the docking run terminates. Because of the limit of 300 poses, the clustering process is fast. The members of the pool are replaced by the new cluster representatives found (limited by the **Max number of poses returned** option).

The clustering procedure works as follows:

1. The pool of solutions is sorted according to energy scores (starting with the best-scoring pose).
2. The first member of the sorted pool of solutions is added to the first initial cluster and the member is assigned to be the cluster representative.
3. The remainder of the pool members are added to the most similar cluster available (using the common RMSD measure) if and only if the RMSD between the representative of the most similar cluster and the member is below a user-specified RMSD threshold. Otherwise, a new cluster is created and the member is assigned to be the cluster representative.
4. The clustering procedure is terminated when the total number of clusters created exceeds **Max number of poses returned** (user-defined

parameter) or when all members of the pool have been assigned to a cluster.

5. When the cluster procedure has terminated, the set of representatives (one from each cluster) is returned.

23 Appendix VI: Supported File Formats

MVD accepts the following molecular structure formats:

- PDB (Protein Data Bank). Supported file extensions: *pdb/ent*.
- Mol2 (Sybyl Mol2 format). Supported file extensions: *mol2*.
- SDF (MDL format). Supported file extensions: *sdf/sd* (for multiple structures) and *mol/mdl* (for a single molecular structure).

Currently, the following information is ignored during import of molecular structures:

- Lone pairs and dummy atoms (all file formats).
- When alternative atoms are reported, only the first alternative is used. The remainder is ignored (all file formats). If one of the other alternatives should be used, change the order of occurrence in the file before import.
- CONNECT records (PDB format).
- SUBSTRUCTURE records are ignored during import but created when structures are exported (Mol2 format).

Notice: Although extensive testing and validation of the import and export of these file formats have been conducted, parsing errors may occur. Compliance with the file format standards/protocols will reduce parsing problems significantly. The import/export routines used have been extended to handle deviations from the file format protocols, but parsing errors may still occur. Found parsing errors can be reported (contact Technical Support or send email to support@clcbio.com).

Additionally, Molegro Virtual Docker uses its own MVDML file format. MVDML is a shorthand notation for *Molegro Virtual Docker Markup Language* and is an

XML-based file format. In general, MVDML can be used to store the following information:

- Molecular structures (atom coordinates, atom types, partial charges, bond orders, hybridization states, ...)
- Constraints (location, type, and constraint parameters)
- Search space (center and radius)
- State information (workspace properties, ...)
- Cavities (location, cavity grid points)
- Camera settings (position and angle)
- Visualization settings (e.g. style and color of molecules, rendering options, hydrogen bonds and electrostatic interactions. See description of Visualization Settings dialog for an overview of all settings).

Notice: Purely graphical objects (e.g. labels, interactions, annotations, backbones, and surfaces) are not saved.

24 Appendix VII: Automatic Preparation

The principles behind automatic preparation in MVD are described below.

Aromaticity

- All rings (closed loops) are identified.
- These rings are 'weeded out', until a 'smallest subset' (capable of covering all ring bonds) remains.
- These rings are considered aromatic if:
  1) For 5-cycles: the mean torsion angle is less then 9.5°
  2) For 6-cycles: the mean torsion angle is less then 12°
- If the aromatic ring contains an atom which has out-of-plane bonds, it is degraded to be non-aromatic.

Notice that this is only a geometrical check for aromacity. It does not include more advanced checks such as Hückel's rule, and may fail on overlapping ring systems.

Assign Hybridization

- All atoms with average bond angles > 155°, are marked as SP1
- All atoms with average bond angles > 115°, are marked as SP2
- All remaining atoms are marked SP3.
- All atoms part of aromatic rings are marked as SP2.
- Ensure that if an atom is SP2 or SP, it must be connected to another SP or SP2 or a terminal atom. Otherwise the atom is degraded (i.e. SP2 -> SP3)

- Lastly the geometry surrounding a SP2 atom should be planar, otherwise it is degraded to SP3.

Bond Order

- All atom bonds are set to 'unknown'. All implicit hydrogens are set to '-1'.
- All bonds to SP3 atoms are set to 'single' order.
- Next, a template file containing standard chemical motifs (-POO-, C(NH2)(NH2), ...) is processed. The templates are located in the file: `\misc\data\preparationTemplates.xml`
- All unset SP2-SP2 bonds involved in a planar geometry (less than 10 degrees) are set to 'double'.
- Next all SP2 atoms are checked to see if a double bond to a neighbour atom is possible. If several atom bonds are possible, the atom with highest electro negativity is chosen. If this still results in several possibilities, the atom closest to the current one will be chosen.

25 Appendix VIII: Third Party Copyrights

MD5

MVD uses a derivate of the MD5 hash algorithm "RSA Data Security, Inc. MD5 Message-Digest Algorithm", under the following license:

You may use this software free of any charge, but without any warranty or implied warranty, provided that you follow the terms of the original RSA copyright, listed below.

Original RSA Data Security, Inc. Copyright notice

Copyright (C) 1991-2, RSA Data Security, Inc. Created 1991. All rights reserved.

License to copy and use this software is granted provided that it is identified as the "RSA Data Security, Inc. MD5 Message-Digest Algorithm" in all material mentioning or referencing this software or this function. License is also granted to make and use derivative works provided that such works are identified as "derived from the RSA Data Security, Inc. MD5 Message-Digest Algorithm" in all material mentioning or referencing the derived work. RSA Data Security, Inc. makes no representations concerning either the merchantability of this software or the suitability of this software for any particular purpose. It is provided "as is" without express or implied warranty of any kind. These notices must be retained in any copies of any part of this documentation and/or software.

Icons

The icon set used in MVD is taken from:

The Tango Icon Library: http://tango.freedesktop.org/Tango_Desktop_Project

They are released under the 'Creative Commons Share-Alike license': http://creativecommons.org/licenses/by-sa/2.5/

26 Appendix IX: Keyboard Shortcuts

The following list contains the keyboard shortcuts available in MVD. On Mac OS X, the CTRL key is replaced by the command key.

- CTRL-O Import Molecules
- CTRL-SHIFT-O Open Workspace
- CTRL-SHIFT-C Clear Workspace
- CTRL-S Save Workspace
- CTRL-F Toggle full screen
- CTRL-H Toggle dockable windows
- CTRL-C Toggle Cofactors category on/off
- CTRL-L Toggle Ligands category on/off
- CTRL-P Toggle Proteins category on/off
- CTRL-W Toggle Water category on/off
- CTRL-Q Quit MVD
- CTRL-1 to 8 Invoke misc. visualization views
- F1 to F9 Invoke misc. dialogs

Notice: Some of the shortcuts can be modified from the **Macro and Menu Editor** and additional shortcuts can be defined for macro commands.

27 Appendix X: Console and Macro Commands

When using the **Macro and Menu Editor**, or entering commands in the console, the following commands can be used.

Notice: Some commands require a *molecule target*: these can be described using the following syntax:

Ligand[0] – the ligand with ID 0.

Ligand[4,5,6] – the Ligands with IDs 4,5 and 6. Multiple IDs are separated by comma.

Ligands – All ligands. By using the plural form of a category, all molecules in it are selected. The categories are: Pose, Cofactor, Protein, Water, Ligand.

Poses;Cofactors;Proteins;Ligands;Water[0]– All Poses, Cofactors, Proteins, Ligands and the first Water molecule. Multiple targets can be concatenated using a semi-colon.

Notice: The IDs of molecules are based on the order of occurrence in the corresponding **Workspace Explorer** category. For instance, ligand molecules listed in the **Ligands** category, begins with index 0 with increments of 1 (i.e. 0,1,2,3,...). If molecules are removed from the workspace, the IDs of the molecules are changed to follow the new order of occurrence in the list.

Command	Description
EXPORT [moleculetarget]	Export as Mol2 or PDB. A File export dialog is opened for selection of a filename.
SURFACEDIALOG	Shows the Surface dialog.
PREPAREDIALOG	Shows the Preparation wizard.
DISTANCECONSTRAINT	Shows the Distance constraint dialog.
LABELDIALOG	Shows the Label dialog.
DOCKINGWIZARD	Shows the Docking Wizard.
GETPDB <key>	Downloads PDB with 'key' (4 letter code) from the Protein Data Bank.
ALIGN [MoleculeTarget1] [id1][id2] [id3] [MoleculeTarget2] [id1][id2] [id3]	Aligns atom id1,id2,id3 in MoleculeTarget1 with atom id1,id2,id3 in MoleculeTarget2.
SHOW CATEGORY <category> HIDE CATEGORY <category>	Shows or hides Workspace Explorer category with given name: i.e. SHOW CATEGORY water
CD	Print current directory.
DIR	Shows dir of MVDML files in current directory.
MKDIR <directory>	Make a new directory named 'directory'.
RM <directory>	Remove directory named 'directory'.
PREV	Loads previous MVDML file in current directory.
NEXT	Loads next MVDML file in current directory.
RMSD	Invokes RMSD dialog.
CAV	Invokes Cavity detection dialog.
SELECT ID <id> SELECT ATOM <x y z> SELECT RESIDUE <id> SELECT RESIDUEID <id>	Selection of objects: 'SELECT ID' selects all atoms with id = 'id'. 'SELECT ATOM' selects closest atom to specified x, y, z position.

	'SELECT RESIDUE' selects residue with residue index = 'id'. 'SELECT RESIDUEID' selects residue with internal residue index = 'id'.
SEED [number]	Sets random seed. It shows the current random seed if called without arguments.
STATUS	Shows info about the objects in the workspace and Visualization Window. Loaded modules are also listed.
SAVE [filename]	Saves a MVDML file. Do not include extension in filename.
LOAD [filename]	Loads a MVDML file. Do not include extension in filename.
CLS	Clears console log.
CLEAR [workspace|selection]	'CLEAR workspace' removes all items in the current workspace. 'CLEAR selection' clears current selection.
HIDE [hydrogens | labels]	Hides either hydrogens or labels.
SHOW [hydrogens | labels]	Shows either hydrogens or labels.
FITTOSCREEN	Fit all molecules in the visualization window.
ADDLABEL	Used for labeling objects. This command is described in detail in the paragraph below. *Notice: It is much easier to use the Label dialog in the GUI*.
GUI Commands	
SLAB [near] [far]	Creates a slab (slicing) of the 3D world. *Notice: The Clipping Planes dialog is easier to use.*
QUALITY [value]	Sets OpenGL rendering quality from 0 to 10.
LIGHT [number] [on|off] [ambient] [diffuse] [specular] {[x] [y] [z]}	Sets OpenGL light sources.
FOG LINEAR [near] [far] FOG [EXP|EXP2] [exponent]	Sets OpenGL fog.

FOG OFF	
COLOR [protein|pose|ligand|water|cofactor] [fixed|cpk|hbond|hbond2|interaction|interaction2] {r g b}	Sets the color style of specified object. For more information about color styles, see the 'Visualization Settings' dialog section.
STYLE [protein|pose|ligand|water|cofactor] [vdw, fixed, stick, wireframe, none] atomScale bondScale lineWidth	Sets the visualization style of specified object. The last parameter lineWidth is only used in wireframe mode, and is the line width in pixels. For more information about graphical styles, see the 'Visualization Settings' dialog section.
PROJECTION [perspective|orthogonal] angle	Determines perspective projection mode. Angle is the field-of-view angle for perspective projection. For more information see the 'Visualization Settings' dialog section.
BACKGROUNDCOLOR r g b	Sets the background color
LABELCOLOR r g b	Sets the label color
CAVITYCOLOR r g b	Sets the cavity color
REBUILD	Rebuilds all objects in the Visualizer Window. This command is necessary to call after the visualization styles or coloring schemes have been updated. Otherwise graphical changes will not be reflected in the GUI.

The addlabel command works in the following way: it scans the input-string for known variables (like ID, HYB, ELE - see below) and replaces them with their value. That is, the command 'label bond bond_number:id' will add a label of type 'bond number x' to every bond (underscores are replaced with spaces). To clear all labels use 'label' without any argument.

Variable	**Description**
Atom labels. Syntax: '**Addlabel *string***'	
ID	Internal atom index
Type	Hydrogen bond type: non-polar, acceptor, donor, both. The HBOND variable below is probably of more use.
PC	Partial Charge.

PC!	PC! ignores atoms with no partial charge.
HYB HYB!	Hybridization. HYB! only displays hybridization for atoms with other hybridizations than SP3 or unknown.
SP2	Labels SP2 hybridized atoms
SYM	Element symbol. (H, C, N, ...)
ELE	Element number.
IH	Number of implicit hydrogens.
HBOND HBOND!	Hydrogen bond type shown as : D, A, D+A, - (non-polar) HBOND! ignores non-polar atoms.
ETOT	Shows the total energy of the atom. This requires that the energy has been evaluated using the 'eval' command.
PDB Atom Name	Show PDB Atom Name
PDB Index	Show PDB atom index
Bond labels. Syntax: '**Addlabel bond *string***'	
ID	Internal bond index.
Type	Bond order: single, double, triple, aromatic, ...
ETOT	Shows the total energy of the bond. This requires that the energy has been evaluated using the 'eval' command.
Residue Labels. Syntax: '**Addlabel residue *string***'	
ID	Internal residue index
LONGNAME	Full residue name ('histidine', 'cysteine', ...)
NAME	3-letter abbreviation ('HIS', 'CYS', ...)
LETTER	1-letter abbreviation.

28 Appendix XI: Script Commands

This appendix describes all the script commands that are available in MVD.

Some script commands require a *molecule target*: these can be described using the following syntax:

Ligand[0] – the ligand with ID 0.

Ligand[4,5,6] – the Ligands with IDs 4,5 and 6. Multiple IDs are separated by comma.

Ligand[50-60] – the Ligands with IDs from 50 to 60 (both included). Ligand ranges are specified by a "-".

Ligands – All ligands. By using the plural form of a category, all molecules in it are selected. The categories are: Pose, Cofactor, Protein, Water, Ligand.

Poses;Cofactors;Proteins;Ligands;Waters – All Poses, Cofactors, Proteins, Ligands and all Water molecules. Multiple targets can be concatenated using a semi-colon.

All – imports all structures.

Notice: The IDs of molecules are defined by their order of occurrence in the workspace. All indices are zero-based, meaning that the first ligand will have index 0, the second index 1, and so forth.

28.1 List of Script Commands Available

Comments in MVD script files

It is possible to add comments to MVD script files using either // for a one line comment or /* */ to span more line.

Notice: Currently, it is not possible to add comments after script commands.

Examples:

```
// This is a one line comment
```

```
/* This is a comment spanning more than one line which can be useful when
describing what is going on */
```

CD <path>

Changes the current working directory to the given path.

INFO <output>

Writes “output” to the console.

Can be useful for debugging loops.

Example:

```
INFO Variable a is $a
// Outputs the value of “$a”
```

CUDADEVICE <id>

Sets the active CUDA device “id” (see Section 6.5 for more details).

IMPORT <targets> FROM <file>

The IMPORT command reads molecular data from either PDB, ENT, Mol2, Mol, SDF, SD files.

<targets> is the usual syntax for specifying the molecules to import.

Notice:

- Files imported using the IMPORT command are always parsed using the currently set parser-settings (see PARSERSETTINGS command) and prepared using the currently set preparation-settings (see PREPARE command).
- Files are always appended to the workspace. (The workspace is not cleared). You can clear the workspace using the NEW command.
- If you want to import MVDML files, use the LOAD command.
- If a complete file path is not specified, the current working directory is used to search for the files (see the CD command).
- The importer is able to read UTF-8 or UTF-16 Unicode encoded files. It is also able to read 8-bit Local encoded files, but will not parse special national characters correctly. If errors are encountered with special characters (for instance in the name of the ligands), try converting the files to Unicode.

Examples:

```
IMPORT Ligand[1,3] FROM testdock.mol2
```

```
IMPORT All FROM testdock.mol2
```

```
IMPORT Proteins;Waters;Cofactors FROM 1hvr.pdb
```

PREPARE <settings-string>

Determines how molecules imported using the IMPORT command are prepared.

The settings-string is composed of semi-colon separated pairs of a preparation-type and its preparation scheme:

Preparation Types: **Bonds**, **BondOrders**, **Hydrogens**, **Charges**, or **TorsionTrees**

Preparation Schemes: **IfMissing**, **Always**, **Never**, or **Remove**

The default setting is:

```
PREPARE Bonds=IfMissing;BondOrders=IfMissing;Hydrogens=IfMissing;Charges=
Always; TorsionTrees=Always
```

It is not necessary to specify all of the PREPARE settings. If only some of them are specified the default parameters will be used for the remainder.

Examples:

```
PREPARE Bonds=Always
// Ensures that we use the built-in algorithm to determine atom connectivity.
```

```
PREPARE Charges=IfMissing
// Uses the charges from the molecular input file (default is to overwrite
them).
```

LOAD <mvdml-filename>

Loads a workspace from a MVDML file.

Notice that this command will replace the current workspace.

No preparation is done on the workspace, since it is assumed that files saved in MVDML format are prepared already.

Notice: LOAD clears the current workspace (if any).

SAVE <mvdml-filename>

Save the current workspace as a MVDML file.

All molecular structures in the workspace are saved.

EXIT

Causes the MVD process to terminate.

This can be useful if running several docking simulations of different proteins automated from a scripting language (i.e. using the Python wrapper.)

Do not use this command when parsing a text-file script as it will terminate the script and not parse anything after the EXIT command.

DOCK <molecules>

The DOCK command initiates the docking process. <molecules> is a list of ligands (notice only ligands are allowed here) to be docked. <molecules> is specified in the usual target format.

The settings for the docking can be modified using the DOCKSETTINGS command. The docking scoring function and search algorithm can be modified using the EVALUATOR and OPTIMIZER commands.

It is also possible to specify a *Data Source* using bracket syntax:

```
DOCK [File=/molecules/test.sdf]
```

See the Data Source chapter for more information. Notice that data source parser is able to read UTF-8 or UTF-16 Unicode encoded files. It is also able to read 8-bit Local encoded files, but will not parse special national characters correctly. If errors are encountered with special characters (for instance in the name of the ligands), try converting the files to Unicode.

Examples:

```
DOCK Ligand[50-60]
// Docks ligand from number 50 to number 60 (both included) in the current
workspace
```

```
DOCK Ligand[0]
// Docks first ligand in the current workspace
```

```
DOCK Ligands
// Docks ALL ligands in workspace
```

EVALUATOR <initstring>

Sets the settings for the evaluator (the docking score function).

There is normally no need to change these.

The **<initstring>** is semi-colon separated string of parameter-value pairs.

The following parameters are available. Their default setting is marked in bold face:

General parameters available to MolDock Score and Plants Score:

cropdistance [double]. Determines whether the protein should be cropped (meaning protein atoms outside a given distance is not taken into account). If crop distance is 0 (the default settings) the size of the active search space is used. For other values, the crop distance is defined from the center of the current reference ligand. Crop distance is measured in Ångstrom. If crop distance is negative, all atoms in the protein will be taken into account. Notice that the docking duration increases with the number of atoms. It is advised to keep the default settings of 0.

tabuclustering = [enabled,rmsd-threshold,score-penalty,rmsd-evalution-mode].

- 'Enabled' turns Tabu-clustering on or off. The possible values are [true| **false**].
- 'rmsd-threshold' determines how close two poses must be before being punished. It is measured in Ångstrom with a default value of 2.0.
- The 'score-penalty' decides the amount that will added to the score-function to punish poses which are close (in terms of RMSD). The value should be positive and has a default value of 100.
- The 'rmsd-evaluation-mode' is either 'id' or 'automorphisms' depending on whether the RMSD should be calculated using matching ids (the fastest) or by taking all automorphisms of the ligand into account (more accurate, but slower). The default is 'id'. When tabu-clustering is enabled in the Docking Wizard with default settings it creates the following initialization fragment: tabuclustering=true,2,100,id.

DisplaceWater = [true | **false**]. Determines whether displaceable water evaluation should be included or not.

DisplaceWaterReward = [**0.0**-10.0]. The entropy reward for displacing a water molecule (only applies when the DisplaceWater option is enabled).

The following parameters are available to MolDock Score and MolDock Score [Grid]:

EVALUATOR <initstring>

ligandes = [true | **false**]. Determines whether the internal electro static energy of the ligand should be included.

internalhbond = [true | **false**]. Determines whether internal hydrogen bonds in the ligand are allowed.

torsion = first, **mean**, all. Determines how torsion terms are evaluated (if several torsion angles are available for a bond).

sp2sp2bond = [true | **false**]. Determines if sp2-sp2 bonds should be taken into account.

eintra = [**true** | false]. Determines whether ligand self-interaction energy should be taken into account.

skiptorsionterm = [true | **false**]. Determines whether ligand torsions are taken into account.

hbond90 = [**true**|false]. Determines whether hydrogen bonding directionality should be taken into account. *Notice:* The **hbond90** option is not available for the grid evaluator or for the PLANTS scoring functions.

The following parameters are available to PLANTS Score and PLANTS Score [Grid]:

ignorehtors = [true | **false**] toggles whether or not hydrogens should be included when calculating the Tripos torsion potential.

Originalplants = [true | **false**] toggles between original Plants setup (using PLANTS specific binding penalty terms and ignoring entries with 'dummy' Tripos atom types in Tripos torsion potential) and MVD implementation of PLANTS score (using another binding penalty term and including 'dummy' Tripos atom types in Tripos torsion potential). See Appendix II:PLANTS Scoring Function for details about the different binding penalty terms available for the PLANTS scoring function.

The **gridresolution** option is only available to grid-based evaluators:

gridresolution = [double]. Sets the grid spacing, where the grid resolution is specified in Ångstrom.

The **SoftenPotential** option is only available for the MolDock Score [Grid]:

SoftenPotential=[true|**false**]. Default is 'false'

Allows you to soften the potential during docking (adjust the treshold and

EVALUATOR <initstring>

strength for the atomic pairwise potentials).

In order to enable softening the project (MVDML-file) must contain a description of the softened sidechains.

This can be made by choosing "**Docking** | **Setup Sidechain Flexibility**" in the GUI.

The default settings from the Docking Wizard will generate the following evaluator string:

```
EVALUATOR cropdistance=0;hbond90=true
```

Notice: an easy way to generate a suitable initstring is to use the Docking Wizard to generate and save a generated script.

EVALUATORTYPE <type>

The EVALUATORTYPE command set the evaluator (scoring function) used while docking.

<type> is one of the following values:

- **MolDockGrid** – for the grid version of the MolDock evaluator.
- **MolDock** – for the standard version of the MolDock evaluator.
- **PlantsGrid** – for the grid version of the PLANTS evaluator.
- **Plants** – for the standard version of the PLANTS evaluator.
- **Ligand** – for an evaluator only taking the ligands internal energy into acoount (for when docking with templates)

Notice: MolDock is set automatically as the default evaluator.

Example:

```
EVALUATORTYPE MolDockGrid
```

MKDIR <path>

Creates a new directory.

OPTIMIZER <initstring>

Sets the settings for the optimizer (the docking search algorithm).

The **<initstring>** is semi-colon separated string of parameter-value pairs.

The following parameters are available. Their default setting is marked in bold. For more information about the parameters see Appendix III: MolDock Optimizer, Appendix XII: MolDock SE, Appendix XIII: Iterated Simplex, or the Docking Wizard section where some of the parameters are described.

popsize [integer=**50**]. Determines the number of individuals in the population.

cavity [**true** | false]. Determines whether poses should be forced to be in cavities.

randomizeligand [**true** | false]. Determines whether the ligand orientation should be randomized before each docking run.

keepmaxposes [int=**5**]

excludeenergythreshold [double=**10000**]

clusterthreshold [double= **0.0**]

The following parameters are used by the MolDock Optimizer algorithm:

scalingfactor [double=**0.50**].

crossoverrate [double=**0.90**].

offspringstrategy [int=**1**]

earlytermination [double=**0.01**]

terminationscheme [int=**0**]

The following parameters are used by the MolDock SE algorithm:

creationenergythreshold=[double]. Default is **100.0**. Poses are only added to the population if the value is this threshold. Notice that when half of the iterations in the docking run have been used, this threshold is automatically turned off in order to ensure that enough poses are created for the simplex evolution phase.

posegenerator=[int,int,int]. Set the Min, Quick, Max number of tries. Default is **10,10,30**. At each step at least 'min' torsions/translations/rotations are tested and the one giving lowest energy is chosen. If the energy is positive (i.e. because of a clash or unfavorable electrostatic interaction) then additional 'max' positions will be tested. If at one time it has not been possible to construct a component which do not clash, the 'max' tries number is

OPTIMIZER <initstring>

lowered to the 'quick' try value.

simplexsteps=[int (default:300)]. The number of iterations of the Nelder-Mead simplex minimization procedure performed at each step of the MolDock SE algorithm.

simplexdistancefactor=[double]. Default is **1.0**. This factor determines how close the point of the initial simplex will be to the other randomly selected individuals in the population. A factor of 1.0 causes the initial simplex to span the neighbour points exactly, while a factor of 0.5 would correspond to simplex points being created halfway between the individuals chosen for optimization and its randomly chosen neighbours. Notice that a factor less than 1.0 will converge slowly. Typical values should be in the range of 0.95 to 3.0.

recombine=[**true** | false]. Allows for turning off the Simplex Evolution phase.

The following parameters are used by the Iterated Simplex algorithm:

maxsimplexsteps [int=**2000**]. Maximum number of steps in Simplex local search performed for each individual.

simplextolerance [double=**0.01**]. The tolerance threshold used to terminate Simplex local search when refining an individual.

simplextolerancebest [double=**0.0001**]. The tolerance threshold used to terminate Simplex local search when refining the best found individual in the current iteration.

usepheromones [true | **false]**. Allows for turning on adaptive sampling using Ant Colony Optimization.

diversify [true | **false]**. Allows for turning on search diversification strategy (see [KORB 2009] for details).

pbest [double=**0.5**]. Probability of best individual. Used by Min-Max strategy for updating pheromone limits (see [KORB 2009] for details).

evaporationrate [double=**0.15**]. Evaporation rate used to adjust pheromone trails (see [KORB 2009] for details).

iterationsdbupdate [int=**5**]. If the best found solution in the last **iterationsdbupdate** number of iterations has higher energy score than the the best solution found since the last diversification event (diversification solution), the diversification solution is used to update the pheromone trails (see KORB 2009] for details). This setting requires that **diversify=true**.

iterationsgbterminate [int=**-1**]. The algorithm is terminated if the global

OPTIMIZER <initstring>

best found solution has not been improved for the last **iterationsgbterminate** number of iterations. The best found solution found is returned.

The default settings (using the MolDock search algorithm) from the Docking Wizard will generate the following optimizer string:

```
OPTIMIZER cavity=false;popsize=50;scalingfactor=0.50;crossoverrate=0.90;
offspringstrategy=1;terminationscheme=0;earlytermination=0.01;
clusterthreshold=1.00;keepmaxposes=5
```

Another example using the MolDock SE search algorithm:

```
OPTIMIZER
populationsize=50;cavity=true;creationenergythreshold=100;posegenerator=10,10
,30;maxsimplex=750;simplexsteps=300;simplexdistancefactor=1
```

Notice: an easy way to generate a suitable initstring is to use the Docking Wizard to generate and save a generated script.

NEW

Clears the current workspace:

All molecules are removed from the workspace.

PARSERSETTTINGS <initstring>

Determines the settings for the molecular parsers used to import the molecules.

The settings-string is composed of semi-colon separated pairs of a parameter key and its corresponding value. The different parameters are:

breakUnrealisticBonds: if enabled this option will break/ignore unrealistic bonds parsed from the molecular file (for SDF and Mol2 files only). Default value is 'false'

combineMoleculeFragments: if enabled this option will combine molecular fragments (Mol2 substructures or small PDB molecules with same chain ID) instead of importing them as independent molecules. Default value is 'true'

useSybylForHybridization: if enabled, Sybyl atom types will be used to determine hybridization (if they are available during import). Otherwise, the default geometric heuristic is used (see Appendix VII: Automatic Preparation for details).

moleculeNameField: if a text string is specified (e.g. `moleculeNameField=id`), SDF molecules containing a data header with the given name will use the content of this header when naming the molecule instead of using the first line in the SDF header. The first header line will also be used if the file does not contain the specified data header.

The default settings corresponds to the following script command:

```
PARSERSETTINGS
breakUnrealisticBonds=false;combineMoleculeFragments=true;useSybylForHybridiz
ation=true;
```

DOCKSETTTINGS <initstring>

Determines the behavior of the docking engine.

The settings-string is composed of semi-colon separated pairs of a parameter key and its corresponding value. The different parameters are:

maxIterations: the value must be an integer describing the maximum number of iterations by the MolDock engine. The default value is 2000.

runs: the number of runs performed for each ligand. Multiple runs will giver higher docking accuracy. The default number is 1. Typically 5 to 10 runs are recommended.

ignoreSimilarPoses: when running multiple runs, several poses are returned for each ligand. Set this to 'true' to weed out similar poses by clustering according to their RMS deviation. Default value is 'true'

IgnoreSimilarPosesThreshold: This is the RMSD treshold value in Ångstrom for the clustering described above. Default value is 'true'.

MaxPoses: Determines the maximum number of poses returned by the clustering described above. Default value is 5

MinimizeReceptor=[LocalSteps,GlobalSteps]. Default is LocalSteps=0,GlobalSteps=0 corresponding to no minimization. Enables minimization of the proteins in the workspace, after each pose returned by the docking engine. For each residue 'LocalSteps' iterations of energy minimization (using a Nelder-Mead Simplex algorithm) is performed for each residue. After that 'GlobalSteps' iterations are performed on all residues at once (again using the Nelder-Mead Simplex algorithm). Receptor minimization is normally used together with a softening of the potentials and Tabu Clustering. If Receptor minimization is enabled a copy of the minimized receptor configuration is saved together with the pose, for each found solution. The receptor configurations will be saved as 'ligandname.receptorConfiguration' and are most easily inspected using the Pose Organizer.

postMinimize: Perform short energy minimization of final poses found after docking. See Section 6.3 for details. Default value is 'false'

postOptimizeHBonds: Optimize hydrogen donor positions (both for pose and protein target atoms). See Section 6.3 for details. Default value is 'true'

The default settings corresponds to the following script command:

```
DOCKSETTINGS maxIterations=2000;runs=1;ignoreSimilarPoses=true;
IgnoreSimilarPosesThreshold=1.0;MaxPoses=5;postMinimize=false;
poseOptimizeHBonds=true
```

It is not necessary to specify all of the parameters. If only some of them are

DOCKSETTTINGS <initstring>

specified the default parameters will be used for the remainder.

Examples:

```
PREPARE maxIterations=4000
// Use a higher number of iterations
```

```
PREPARE runs=10
// Multiple runs increases the accuracy of the poses found.
```

OPTIMIZERTYPE <type>

The OPTIMIZERTYPE command sets the optimizer (search function) used while docking.

<type> is one of the following values:

- **MSE** – for the MolDock SE algorithm.
- **MolDock** – for the standard MolDock algorithm.
- **Simplex** – for the Iterated Simplex algorithm.

Notice: MolDock SE is automatically set as the default optimizer.

Example:

```
OPTIMIZERTYPE MSE
```

RANDOM <seed>

Sets the seed used by the random number generator.

Normally this is not recommended, since a random seed always is generated on startup, but it can be used to reproduce docking runs, since the seed is always recorded in the docking log.

```
RANDOM 123
// Ensures that the simulation will always return the exact same results.
```

SEARCHSPACE <radius center>

Create a 'searchspace' with a given radius and center position. The center is based on a given molecule (ligand, cofactor, pose) or existing cavity.

Example:

```
SEARCHSPACE radius=12;center=ligand[0]
```

CONSTRAINTS <integer list>

Per default all constraints defined in a MVD workspace are used.

The CONSTRAINTS command enables a subset of the constraints in the workspace. All constraints not specified in the list are not used during the docking run.

To disable all constraints, set “integer list” = “NONE”.

It is possible to specify ranges, or to just enable all constraints by setting “integer list” = “ALL”.

Notice: the numbering of constraints is zero-based, meaning that the first constraint in a workspace will have number 0, the second number 1 and so forth.

Examples:

```
CONSTRAINTS 1,2
/* Enables the second and third constraint in the workspace
   All other constraints are disabled */
```

```
CONSTRAINTS 1,3-5
/* Enables the second, fourth, fifth and sixth constraint in the workspace
   All other constraints are disabled */
```

```
CONSTRAINTS NONE
// Disables all constraints in the workspace
```

```
CONSTRAINTS ALL
// Enables all constraints in the workspace (default behavior)
```

RMSD <targetligand>

The RMSD can be used to set a ligand to compare docking results with.

The Root-Mean-Square-Deviation between heavy atoms will be calculated for all returned poses.

Notice: the ligand used as reference for RMSD calculations must have the same number of heavy atoms as the ligands that are docked, otherwise the RMSD calculation will just return -1.

Examples:

```
LOAD 3PTB.MVDML
RMSD ligand[1]
DOCK
// Docks the ligands in 3PTB.MVDML and calculate their RMSD deviation from
ligand[1], which is the second ligand present in the workspace
```

TEMPLATE parameters

The TEMPLATE command is used to specify parameters for template docking.

The following parameters are available:

strength – the normalization constant for a perfect match to the template. Default value is -500.

useGrid – determines if template force field should be precalculated on a grid. Default is true

gridResolution – the resolution of the template force field grid. Default is 0.4 (measured in Å)

Example:

```
TEMPLATE strength=-500;useGrid=true;gridResolution=0.4
```

ADDWATER <x y z>

The ADDWATER command is used to create a water molecule at the position specified and add it to the current workspace.

Example:

```
AddWater 2.4 1.5 7.8
```

DOWNLOAD <PDB code> AS <filename.pdb>

The DOWNLOAD command can be used to download a PDB file from the Protein Data Bank. The downloaded file will be saved as <filename.pdb>.

The downloaded PDB file is not automatically imported to the current workspace. This should be done using the IMPORT command.

Notice: the <PDB code> is a 4-letter PDB identifier and that the filename should include the pdb file extension.

Moreover, the DOWNLOAD command overwrites existing filenames named <filename.pdb>.

Examples:

```
DOWNLOAD 3ptb AS 3ptb.pdb
IMPORT All FROM 3ptb.pdb
DOCK
```

28.2 Flow Control

MVD also provides a couple of simple commands for controlling the script flow. If more complex execution control is needed consider using the Python wrapper to control to scripting engine.

Notice: The variable system in the script parser is strictly string based which means that the script parser simply substitutes occurrences of variable names with the current value before parsing the string.

Also notice that this means that it is important to be careful when defining variable names and ensure that they do not overlap: e.g. do not define two variables named $PDB and $PDBS since the script parser will substitute part of the variable name $PDBS with the value of $PDB.

FOR <$VAR> IN <VALUELIST>
ENDFOR

The FOR command can be used to iterate though a set of possible values.

The VALUELIST must be a comma separated list of values.

FOR commands can be nested (it is possible to have a FOR command inside another FOR loop).

Variables must start with a "$" identifier.

Example (docking multiple complexes):

```
FOR $PDB IN 3PTB,1HVR,1LIC,1TMN
   // $PDB will be replaced by the appropriate value in the loop
   LOAD C:\BENCHMARK\$PDB.mvdml
   RMSD ligand[0]
   DOCK
   NEW
ENDFOR
```

Example (docking with different population sizes):

```
FOR $popsize IN 10,20,30,40,50
   OPTIMIZER cavity=true;popsize=$popsize;crossoverrate=0.9;
   LOAD C:\BENCHMARK\3PTB.mvdml
   RMSD ligand[0]
   DOCK
   NEW
ENDFOR
```

SET <$VAR> = <VALUE>

The SET command can be used to set a variable to a given value.

Variables must start a "$" identifier.

Example:

```
SET $PDB = 3PTB
LOAD C:\BENCHMARK\$PDB.mvdml
RMSD ligand[0]
DOCK
```

29 Appendix XII: MolDock SE

MolDock SE (simplex evolution) is an alternative search heuristic which can be used together with either the *MolDock* or *MolDock [Grid]* scoring functions.

It is known to perform better on some complexes where the standard MolDock algorithm fails. This is usually the case when the ligand has lots of internal degrees of freedom (many torsion angles).

While other algorithms based on parallel simplex search exist, our implementation has been modified to be suitable for docking (by the inclusion of the pose generation step, and the way the initial simplices are created).

The algorithm works as follows:

Pose Generation

First an initial population of poses is created. The initial number of poses is determined by the 'population size' parameter.

These poses are built incrementally from their rigid root point: The pose generator tests a number of different torsions angles, rotations and translations, evaluates the affected part of the molecule and chooses the value which results in the lowest energy contribution.

The torsion angles are chosen from one of three distributions depending on the hybridization of the atoms the bond connects (either sp2-sp2, sp2-sp3 or sp3-sp3).

If the generated pose has an energy below 'energy threshold' it is accepted into the initial population for the 'simplex evolution' algorithm.

Simplex Evolution

The simplex evolution algorithm performs a combined local / global search on the poses generated by the pose generator. The local search is performed using the Nelder-Mead local search algorithm, but unlike Nelder-Mead's original scheme, the algorithm has been extended to take the position of the other individuals in the population into account. At each iteration a random individual is chosen. The representation of this individual determines the first point of the simplex in the N-dimensional search space. Then N additional individuals are chosen and their representations define the remaining N points of the simplex (a simplex in N dimensions has N+1 points). Notice that 'Neighbour distance factor' parameter determines how much the initial simplex should be enlarged or shrinked (see below).

Usage From GUI

In order to use the search algorithm choose '**Search algorithm** -> **Algorithm** -> **MolDock SE**' from the **Docking Wizard**.

The following parameters can be set:

Max iterations: (default=1500) The number of steps per run. These steps are evenly divided between the pose generator and the simplex evolution algorithm (even though both of these may terminate before the number of iterations has been used).

Max population size: (default=50) The number of individuals in the simplex evolution phase. Notice that this number must be higher than the number of degrees of freedom (7 spatial degrees of freedom plus the number of chosen rotatable torsion bonds).

Pose Generation Parameters

Energy threshold: (default=100.00) Poses are only added to the population if the value is below this threshold. Notice that when half of the iterations in the docking run have been used, this threshold is automatically turned off in order to ensure that enough poses are created for the simplex evolution phase.

Tries: **Min**, **Quick**, **Max**.

At each step at least 'min' torsions/translations/rotations are tested and the one giving lowest energy is chosen. If the energy is positive (i.e. because of a clash or an unfavorable electrostatic interaction) then additional 'max' positions will be tested. If it is not possible to construct a component which do not clash, the 'max' tries number is lowered to the 'quick' try value.

Simplex Evolution Parameters

Max Steps: (default=300). The number of iterations of the Nelder-Mead

simplex minimization procedure performed at each step of the MolDock SE algorithm.

Neighbour distance factor: (default=1.0). This factor determines how close the point of the initial simplex will be to the other randomly selected individuals in the population. A factor of 1.0 causes the initial simplex to span the neighbour points exactly, while a factor of 0.5 would correspond to simplex points being created halfway between the individuals chosen for optimization and its randomly chosen neighbours. Notice that a factor less than 1.0 will converge slowly. Typical values should be in the range of 0.95 to 3.0.

Usage When Scripting

To use the MolDock SE search algorithm, the OPTIMIZERTYPE script command has to be set. Moreover, specific search algorithm parameters are set by the OPTIMIZER script command (see Appendix XI: Script Commands for more details).

30 Appendix XIII: Iterated Simplex

Iterated Simplex is an alternative search heuristic which can be used together with the *MolDock* and *PLANTS* docking scoring functions.

The algorithm works as follows: First an initial population of poses is created (initial number of poses is determined by the **population size** parameter). Afterwards, the following process will be executed until **max iterations** have occurred: Each individual in the population will be refined using the Simplex local search algorithm (also called Nelder-Mead). The Simplex algorithm will run for **maximum steps** or until the fractional difference between the best and worst vertices in the Simplex (w.r.t. the docking scoring function used) is below a given **tolerance**. When all individuals have been refined, the best found individual (named: iteration best solution) will be further refined using the same Simplex algorithm again but with a lower tolerance (**Tolerance (iteration best solution)**). When **max iterations** have occurred the algorithm terminates and returns the best found solution(s).

By enabling the **Constrain poses to cavity** option in the Docking Wizard, the Iterated Simplex algorithm uses a cavity prediction algorithm (introduced in Appendix IV: Cavity Prediction) to constrain predicted conformations (poses) during the search process. More specifically, if a candidate solution is positioned outside the cavity, it is translated so that a randomly chosen ligand atom will be located within the region spanned by the cavity. Naturally, this strategy is only applied if a cavity has been found. If no cavities are reported, the search procedure does not constrain the candidate solutions.

The Iterated Simplex algorithm is generally more robust (w.r.t. reproducing docking results with similar scores) than the MolDock SE and MolDock Optimizer. Therefore, the default number of runs in the Docking Wizard is set to 1. In some cases more runs (e.g. 5) might be necessary to identify good binding modes - in particular when docking very flexible ligands.

Usage From GUI

In order to use the search algorithm choose '**Search algorithm** -> **Algorithm** -> **Iterated Simplex**' from the **Docking Wizard**.

The following parameters can be set:

Max iterations: (default=100) The number of steps per run.

Population size: (default=20) The number of individuals sampled during each iteration of the algorithm.

Maximum Steps: (default=2000). The number of iterations of the Nelder-Mead simplex minimization procedure performed for each individual in the population.

Tolerance: (default=0.01).

Tolerance (iteration best solution): (default=0.0001).

Adaptive Sampling

The Iterated Simplex algorithm can use an adaptive sampling strategy based on Ant Colony Optimization (ACO). The idea in ACO is to use pheromone trails to bias the initialization of individuals towards regions previously resulting in good solutions. The pheromone trails are updated in each iteration of the search algorithm based on the currently best found solution. The parameters **Evaporation rate** and **Probability of best ant (pBest)** are used to control how much the pheromones are modified. For more details about ACO and the parameters, see [KORB 2009].

By default, adaptive sampling is not enabled in MVD since it did not produce better docking results when including pheromone trails (benchmarked on 85 complexes).

Evaporation rate: (default: 0.15)

Probability of best ant (pBest): (default: 0.5)

Usage When Scripting

To use the Iterated Simplex search algorithm, the OPTIMIZERTYPE script command has to be set. Moreover, specific search algorithm parameters are set by the OPTIMIZER script command (see Appendix XI: Script Commands for more details).

31 Appendix XIV: Grid-based Scores

'MolDock Score [Grid]' and 'PLANTS Score [Grid]' are grid based versions of the MolDock Score and Plants Score functions, respectively.

The grid based scoring functions precalculate potential-energy values on an evenly spaced cubic grid in order to speed up calculations. The energy potential is evaluated by using tri-linear interpolation between relevant grid points. The rest of the terms in the Grid based versions (i.e. internal ligand energy contributions and constraint penalties) are identical to the standard version of the scoring functions.

Notice that unlike the standard MolDock Score, the grid version of MolDock Score does not take hydrogen bond directionality into account (hydrogen bonding is determined solely on distance and hydrogen bonding capabilities)

Grids are not stored permanently - they are calculated when needed. (Grid generation is relatively fast. Typically ~15 seconds for the standard settings). Grids will automatically be reused while running docking scripts as long as the target protein does not change.

Notice that large energy grids with high resolution can consume a lot of memory. Grid resolutions of 0.3 Å -0.4 Å will be adequate in most cases. Look out for the estimated memory usage in the Docking Wizard. As a rule of thumb it should never exceed more than half of the physical memory available in the computer. Also notice that if several instances (processes) of MVD is running, each process will need to generate its own grid.

Usage From GUI

In order to use the MolDock Score grid version, select it as the evaluation function in the **Docking Wizard** ('**Scoring Function** -> **Score** -> **MolDock Score [GRID]**').

In order to use the PLANTS Score grid version, select it as the evaluation function in the **Docking Wizard** ('**Scoring Function** -> **Score** -> **PLANTS Score [GRID]**').

Usage When Scripting

To use the grid-based scoring function, the EVALUATORTYPE script command has to be set. Moreover, specific grid parameters are set by the EVALUATOR script command (see Appendix XI: Script Commands for more details).

32 Appendix XVI: References

[THOMSEN 2006] Thomsen, R.; Christensen, M. H. MolDock: A New Technique for High-Accuracy Molecular Docking. *J. Med. Chem.*, 2006, 49(11), 3315-3321.

[CCG] Chemical Computing Group, www.chemcomp.com

[SCHRODINGER] Schrödinger, LLC, www.schrodinger.com

[GEHLHAAR 1995] Gehlhaar, D. K.; Verkhivker, G.; Rejto, P. A.; Fogel, D. B.; Fogel, L. J.; Freer, S. T. Docking Conformationally Flexible Small Molecules Into a Protein Binding Site Through Evolutionary Programming. *Proceedings of the Fourth International Conference on Evolutionary Programming,* 1995, 615-627.

[GEHLHAAR 1998] Gehlhaar, D. K.; Bouzida, D.; Rejto, P. A. Fully Automated And Rapid Flexible Docking of Inhibitors Covalently Bound to Serine Proteases. *Proceedings of the Seventh International Conference on Evolutionary Programming* 1998, 449-461.

[YANG 2004] Yang, J-M.; Chen, C-C. GEMDOCK: A Generic Evolutionary Method for Molecular Docking. *Proteins,* 2004, *55*, 288-304.

[MCDONALD 1994] McDonald, I. K.; Thornton, J. M. Satisfying Hydrogen Bonding Potential in Proteins. *J. Mol. Biol.,* 1994, *238*, 777-793.

[MICHALEWICZ 1992] Michalewicz, Z. *Genetic Algorithms + Data Structures = Evolution Programs*; Springer-Verlag: Berlin, 1992.

[MICHALEWICZ 2000] Michalewicz, Z.; Fogel, D. B. *How to Solve It: Modern Heuristics*; Springer-Verlag: Berlin, 2000.

[STORN 1995] Storn, R.; Price, K. Differential Evolution - A Simple And Efficient Adaptive Scheme for Global Optimization over Continuous Spaces. Tech-report, International Computer Science Institute, Berkley, 1995.

[SHOEMAKE 1992] Shoemake, K. Uniform Random Rotations. In *Graphics Gems III*, 1st ed.; Kirk, D., Ed.; AP Professional (Academic Press); Boston, 1992; pp. 124-132.

[HAYKIN 1999] Haykin, S. Neural Networks: A Comprehensive Foundation. Prentice-Hall, Inc.: New Jersey, 1999.

[SELWOOD 1990] Selwood, D. L.; Livingstone, D. J.; Comley, J. C. W.; O'Dowd, A. B. ; Hudson, A. T.; Jackson, P.; Jandu, K. S.; Rose, V. S.; Stables, J. N. Structure-Activity Relationships of Antifilarial Antimycin Analogues: A Multivariate Pattern Recognition Study, *J. Med. Chem.*, 1990, 33(1), 136-142.

[KORB 2009] Korb, O.; Stutzle, T.; Exner, T. E. Empirical Scoring Functions for Advanced Protein-Ligand Docking with PLANTS, *J. Chem. Inf. Model.*, 2009, 49(1), 84-96.

[CLARK 1989] Clark, M.; Cramer III, R. D.;Opdenbosch, N. Van. Validation of the General Purpose Tripos 5.2 Force Field, *J. Comp. Chem.*, 1989, 10(8), 982-1012.